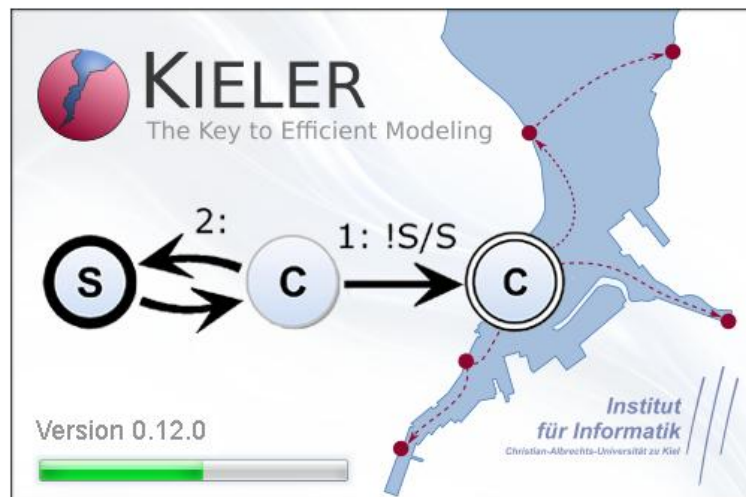
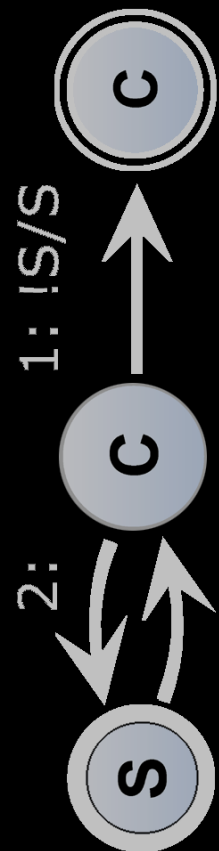




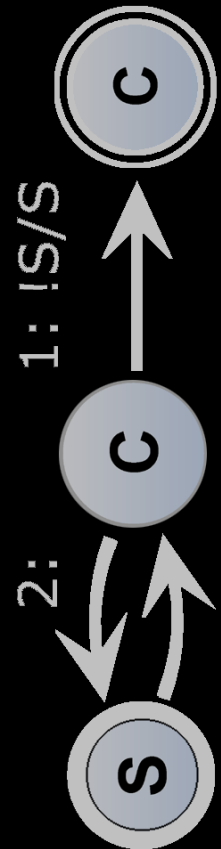
KIELER SCCharts

Tutorial Workshop

Christian Motika, Steven Smyth, Alexander Schulz-Rosengarten, and Reinhard von Hanxleden



SYNCHRON 2016
06. November, Bamberg



SCCharts Intro

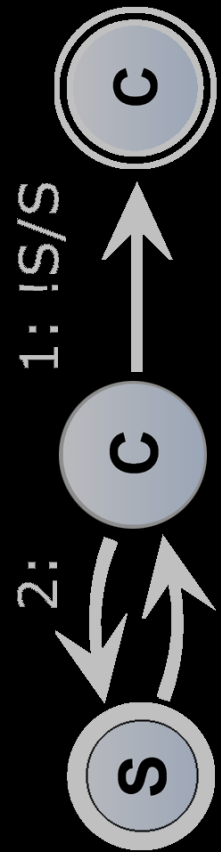
Reinhard von Hanxleden

KIELER SCCharts Tools

Christian Motika

Lego Mindstorms

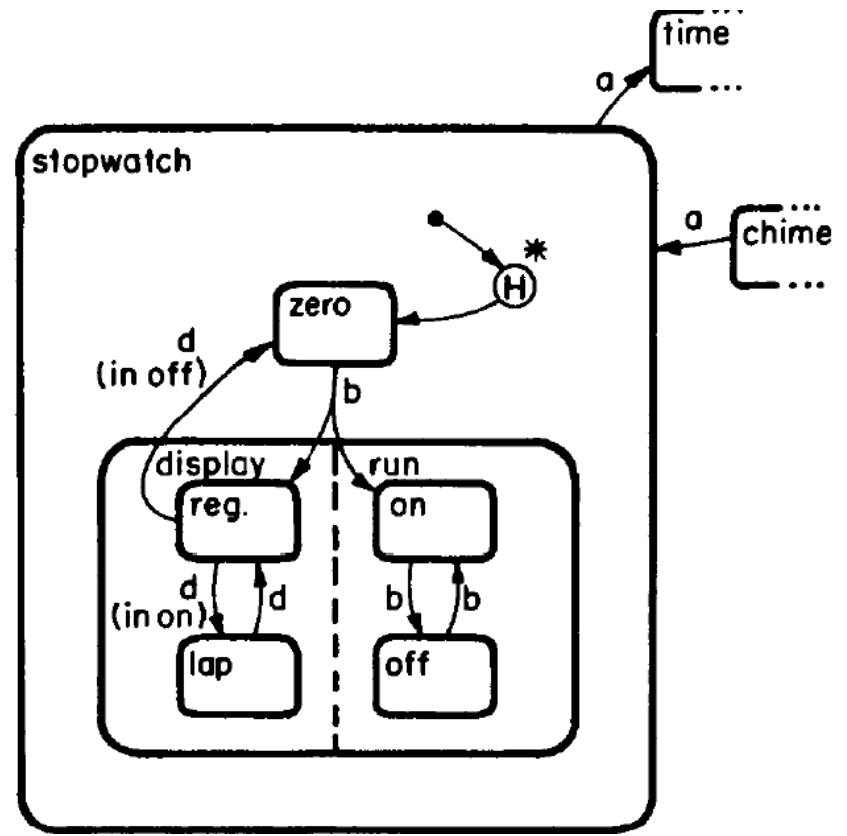
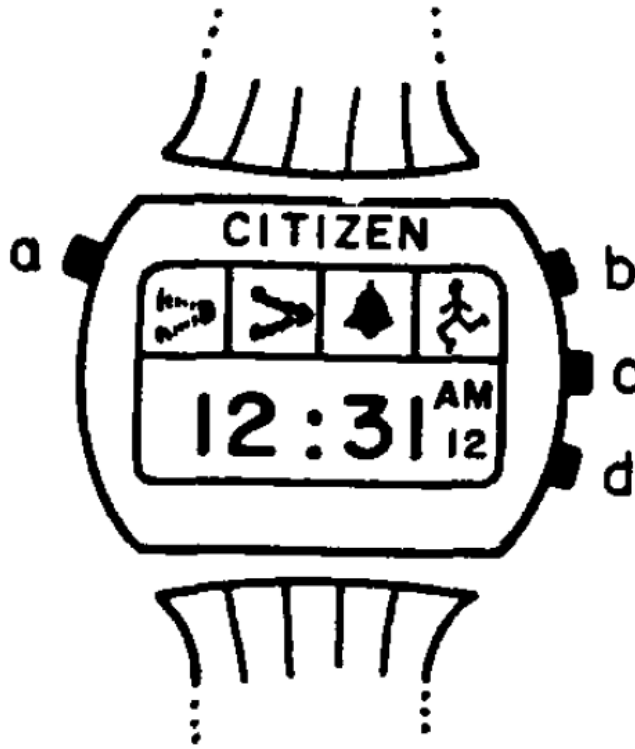
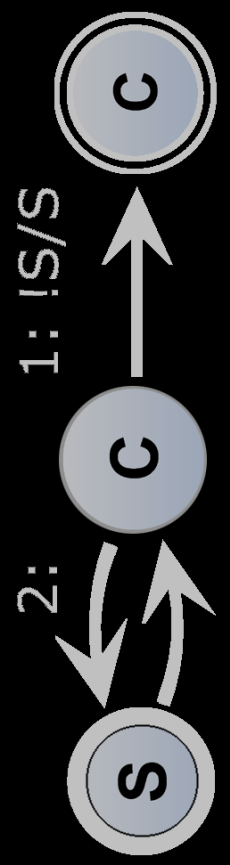
Steven Smyth



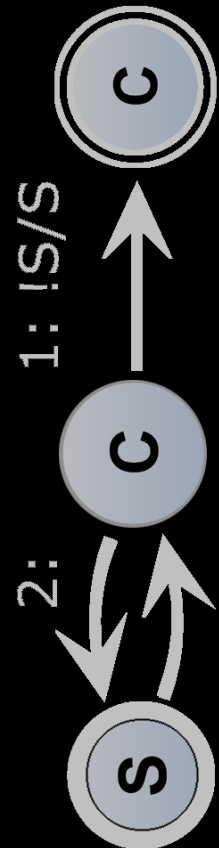
Part I

SCCharts Intro

1980: Statecharts



Harel
Statecharts: A Visual Formalism for Complex Systems
Science of Computer Programming, 1987



28. November – 02. Dezember 1994, Dagstuhl Seminar 9448

Synchronous Languages

Organisatoren

G. Berry, W.P. de Roever, A. Poigné, A. Pnueli

Auskunft zu diesem Dagstuhl Seminar erteilt

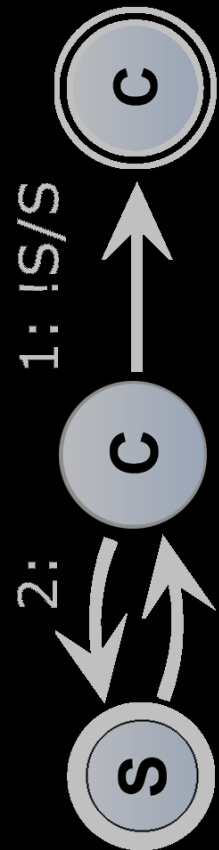
✉ Dagstuhl Service Team

Dokumente

Dagstuhl-Seminar-Report 104

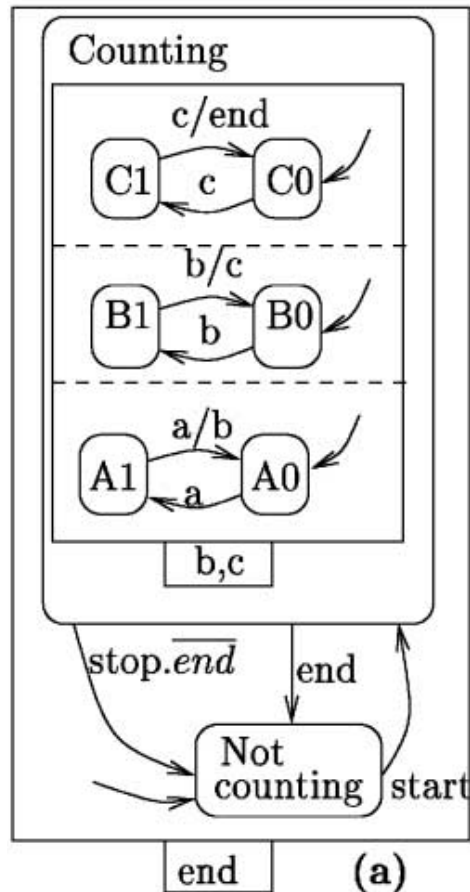
A Historical Upshot

Synchronous Languages such as LUSTRE, ESTEREL and SIGNAL were conceived in the first half of the eighties by mainly French researchers. Independently, Harel & Pnueli worked on an almost synchronous language, STATECHARTS, as part of the STATEMATE system for the specification of real-time embedded systems (mainly concerning software for the aircraft industry). Independently, Ward & Mellor published in 1985 their 3-volume approach "Structured Development of Real-time Systems", containing important synchronous elements.

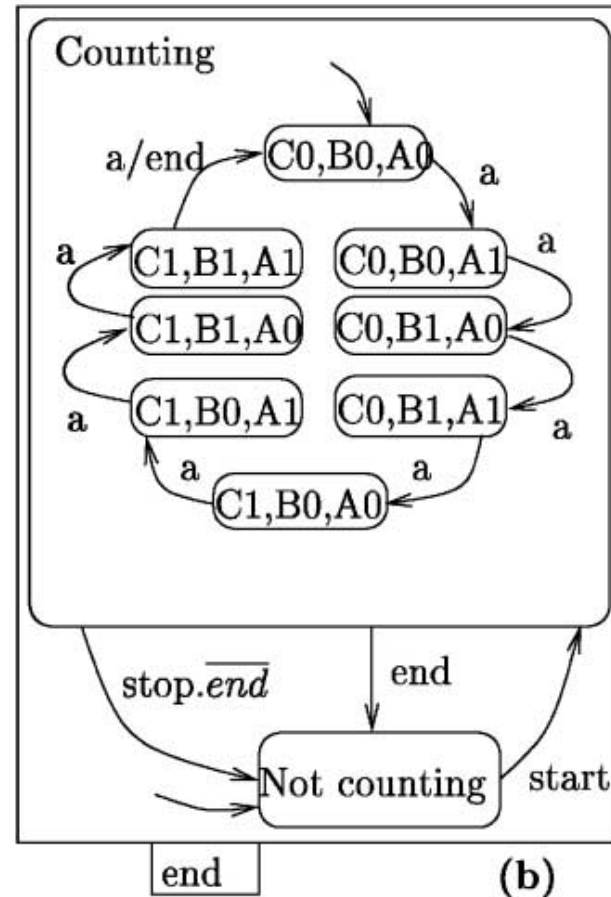


1991: ARGOS

Main1 (a, start, stop) ()



Main2 (a, start, stop) ()



Florence Maraninchi

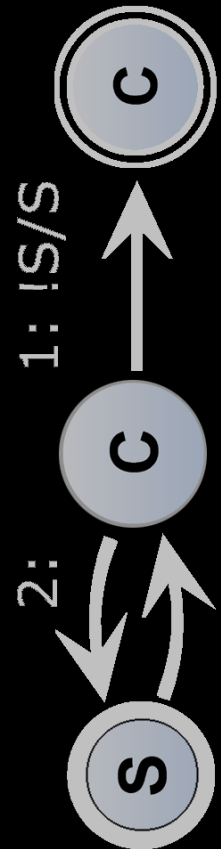
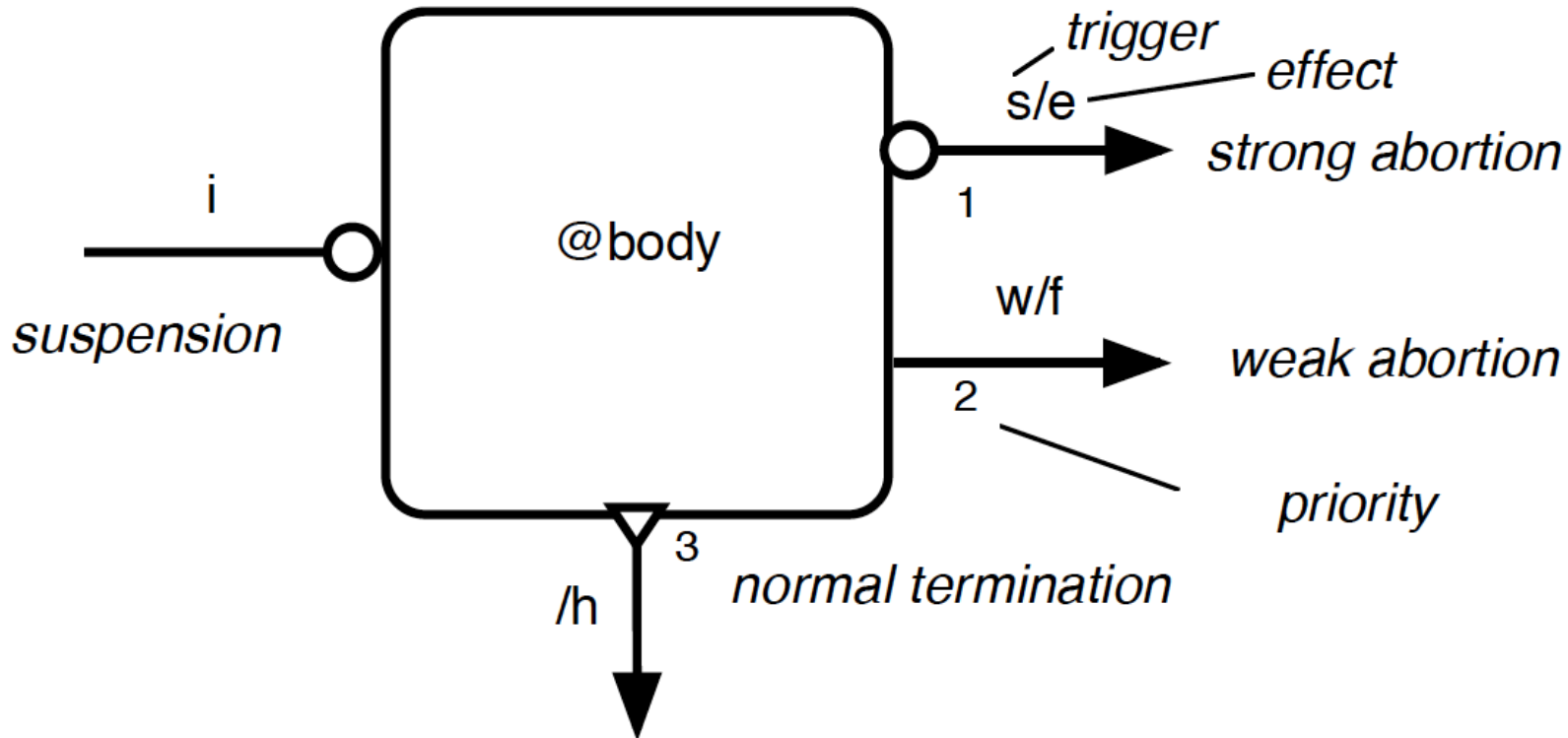
The Argos language: Graphical Representation of Automata and Description of Reactive Systems

IEEE Workshop on Visual Languages, Kobe, Japan, 1991



1995: SyncCharts

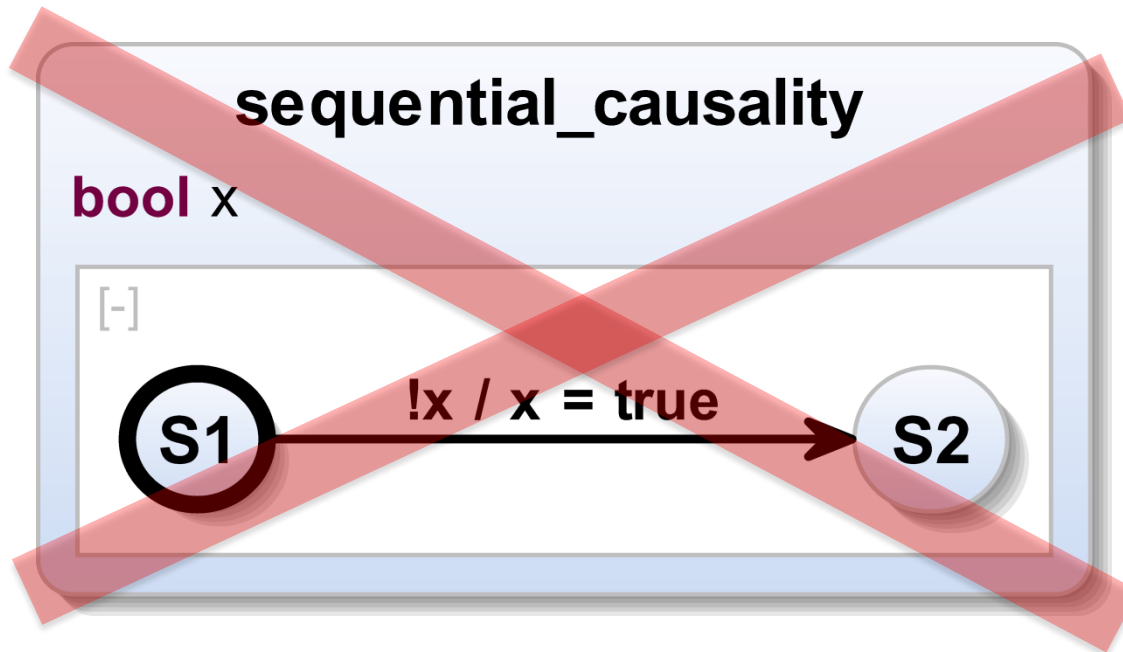
a.k.a. Safe State Machines



Charles André
SyncCharts: A Visual Representation of Reactive Behaviors
Research Report 95-52, I3S, Sophia Antipolis, 1995

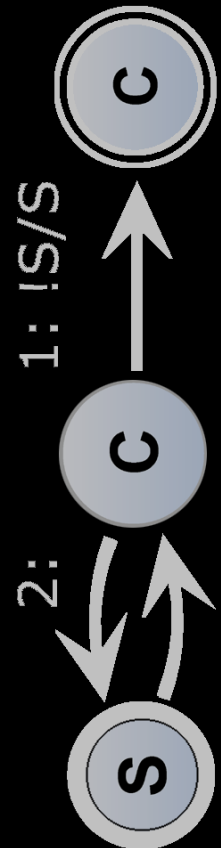


Limitations of Strict Synchrony

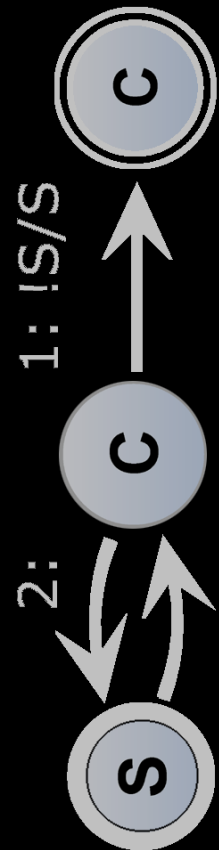


```
if (!x) {
  ...
  x = true;
}
```

Forbidden for shared variables/signals in SyncCharts etc., but allowed in SCCharts



2013: SCCharts



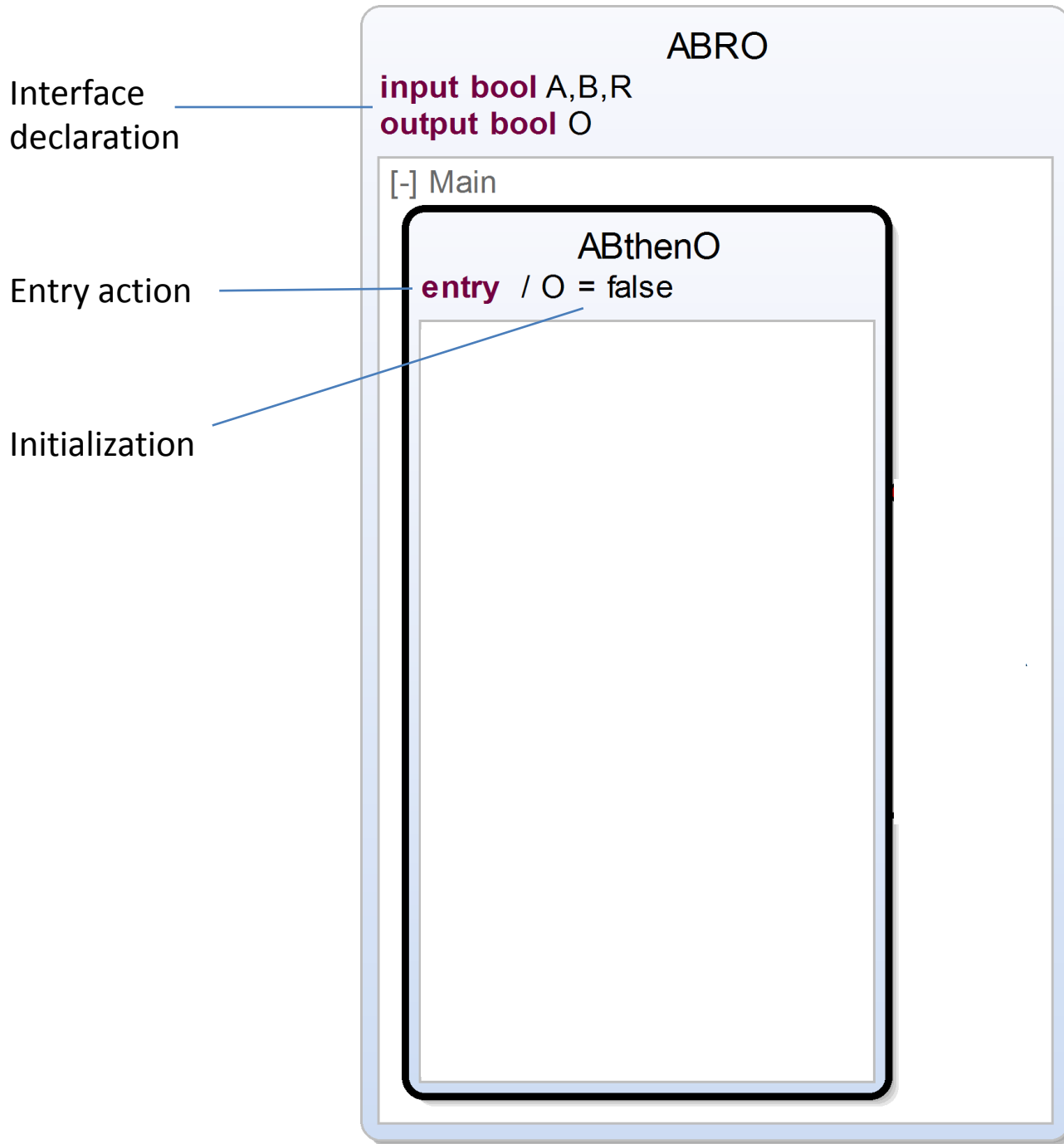
- Based on Sequentially Constructive Model of Computation (SC Moc)
- Collaboration Kiel, Bamberg, Auckland, NI (& friends 😊)
- Funding from industry & DFG (PRETSY 1 / 2)

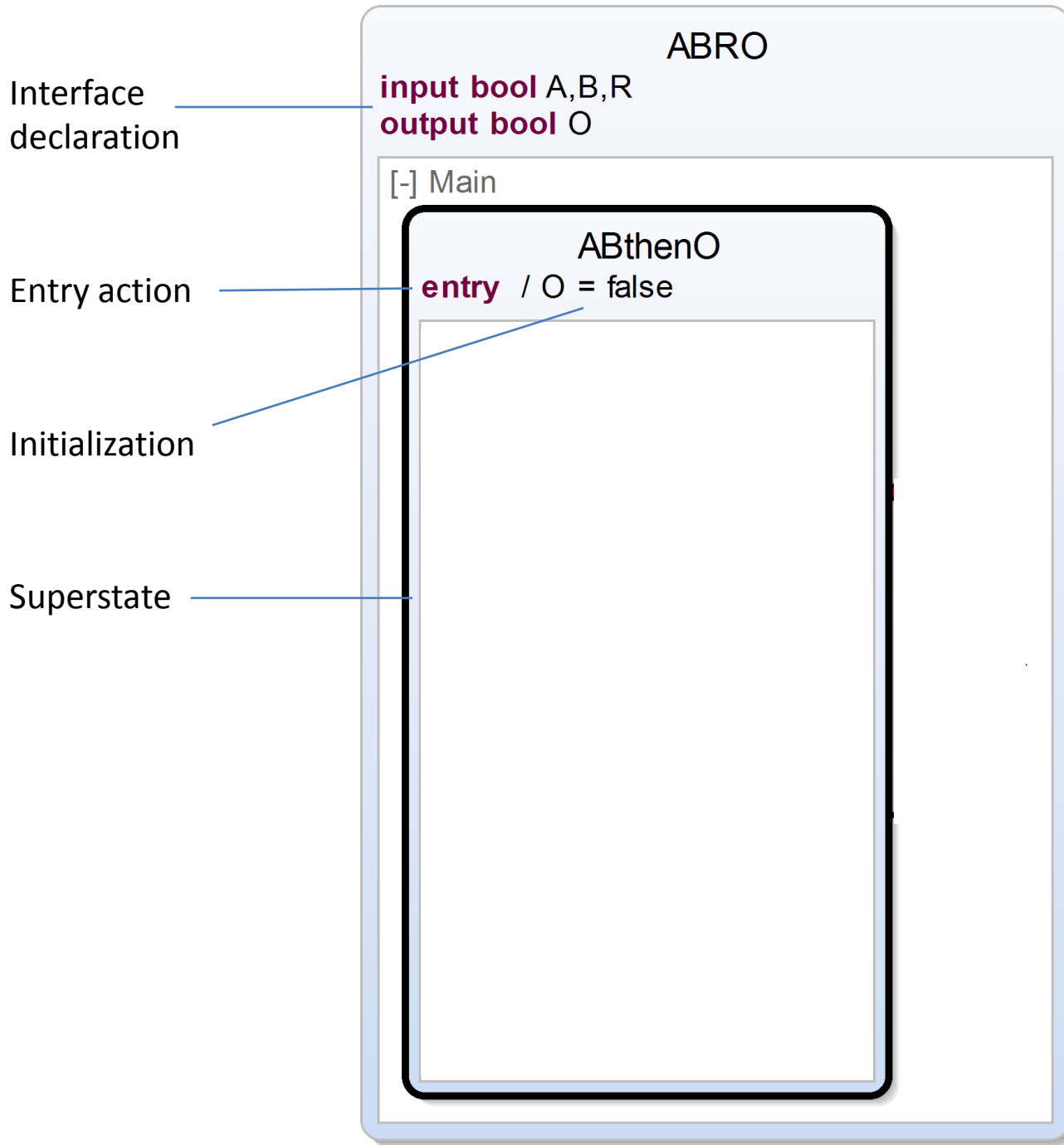
ABRO

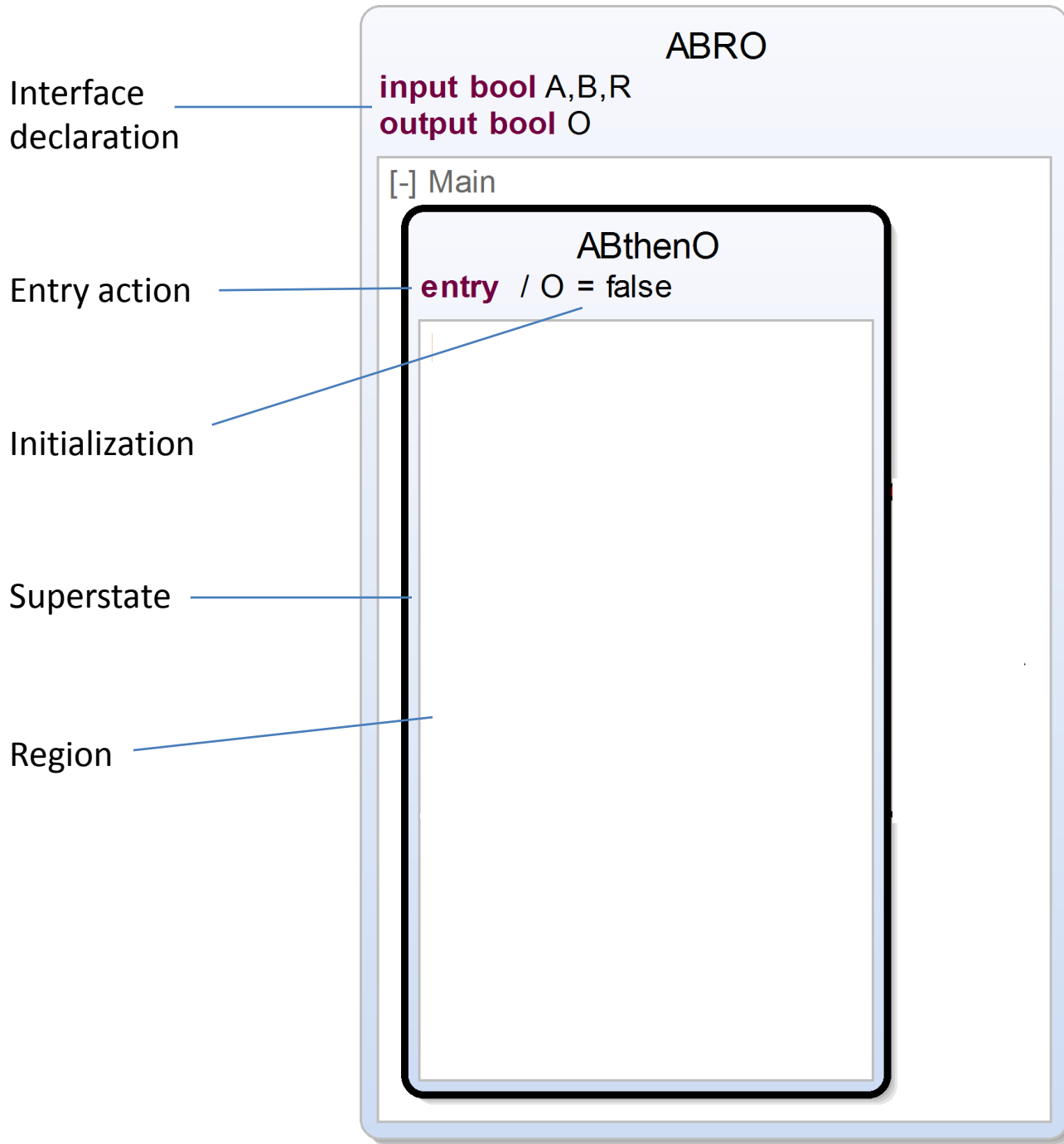
Interface
declaration

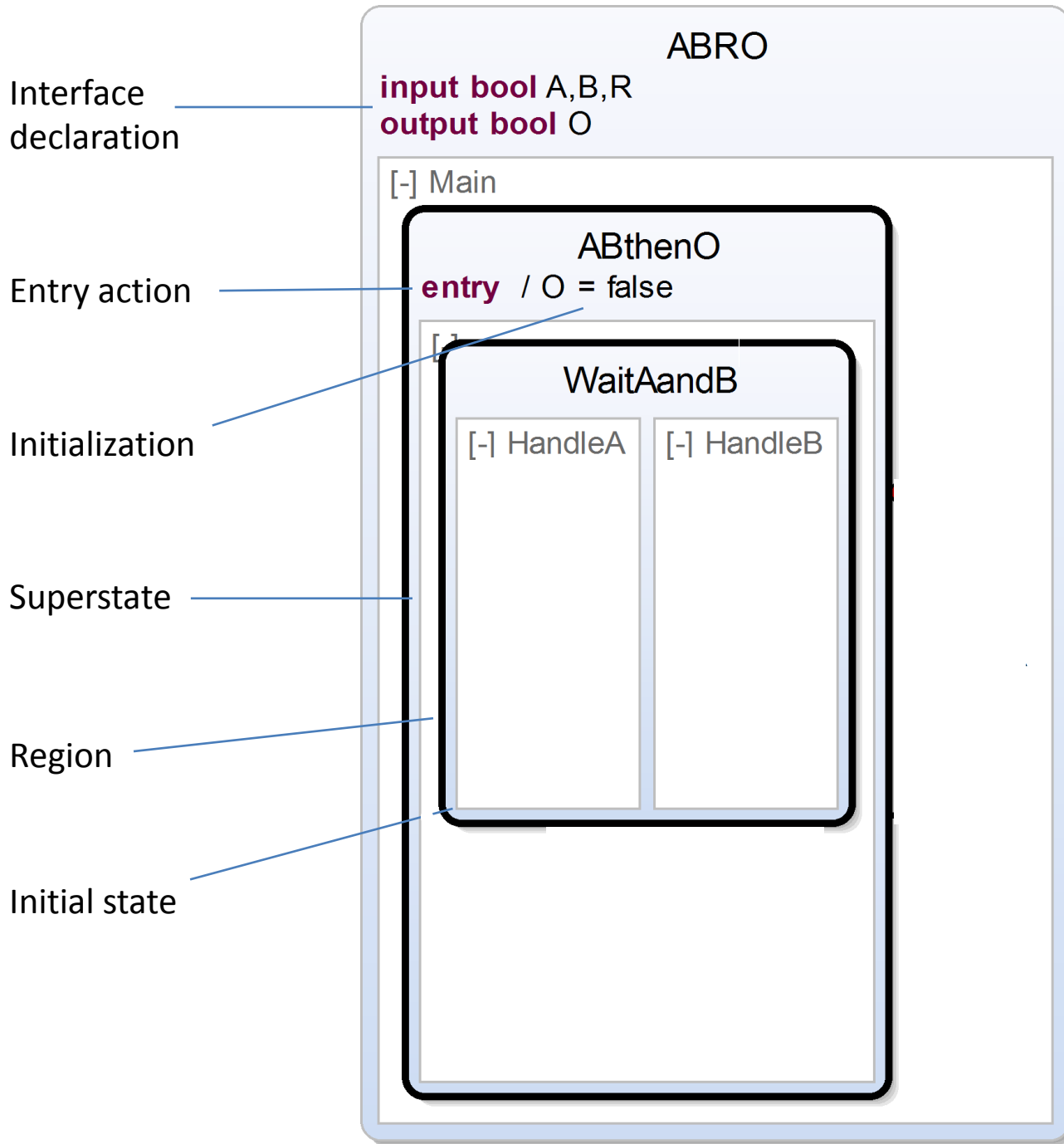
```
input bool A,B,R  
output bool O
```

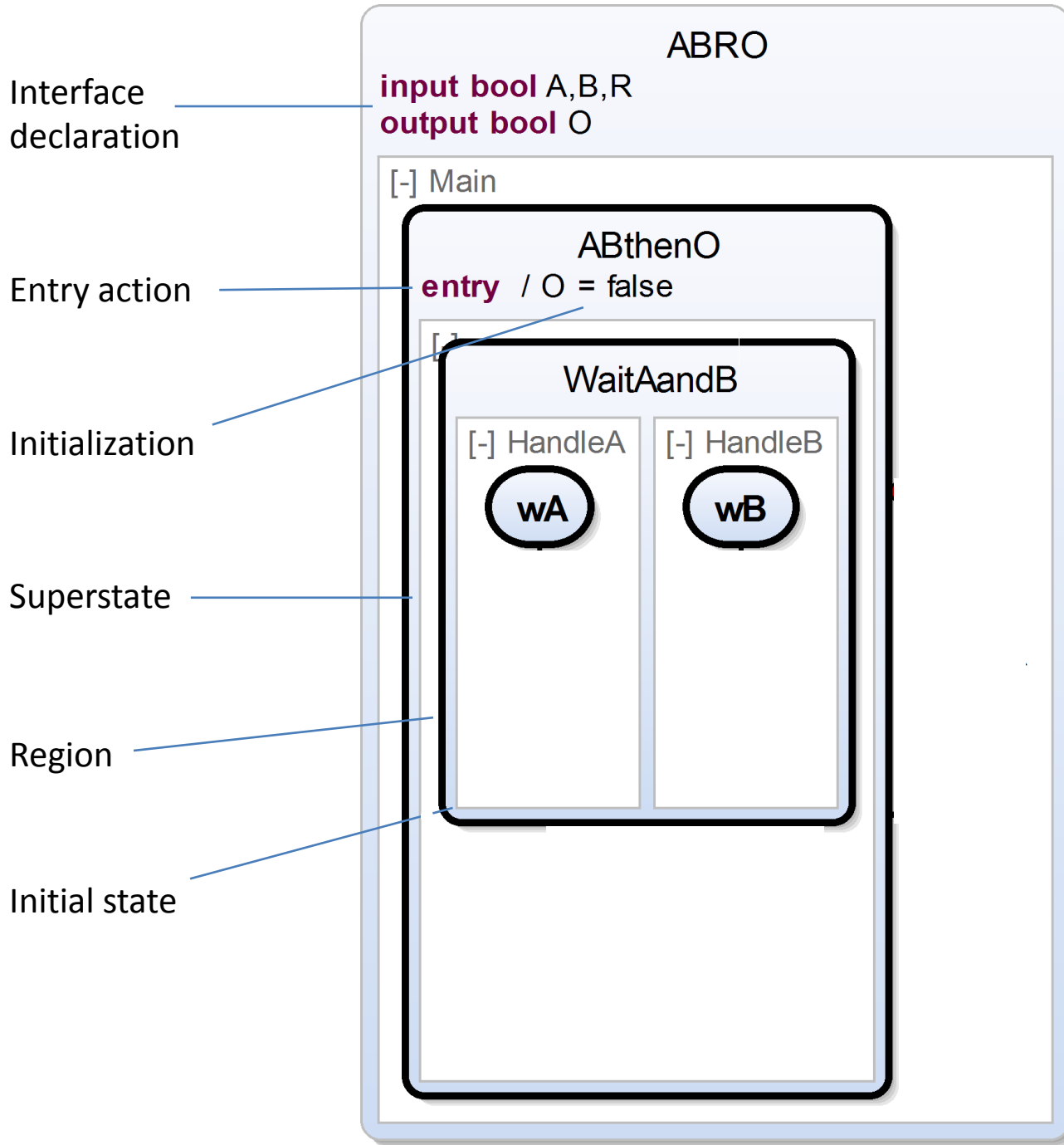
```
[-] Main
```

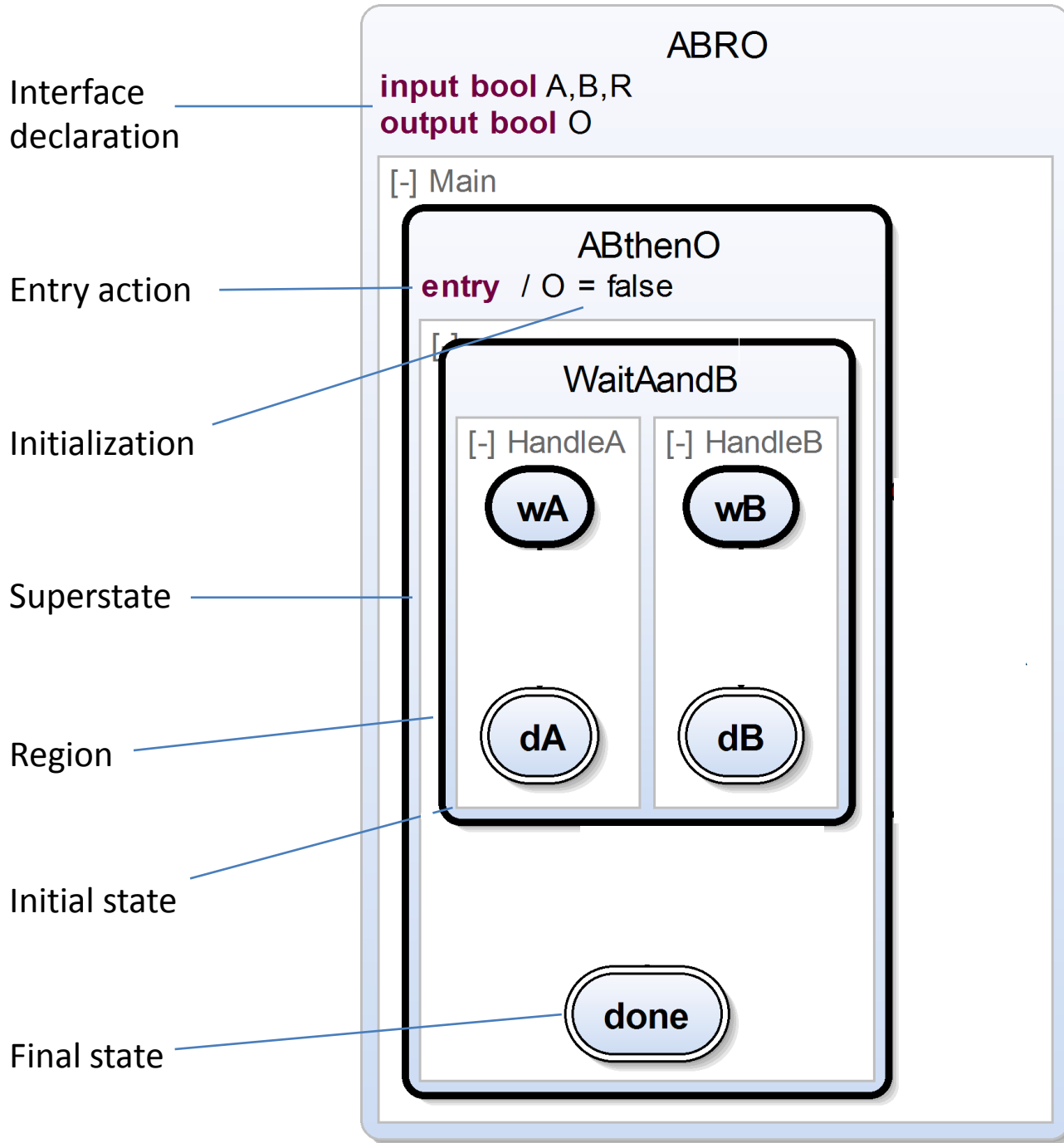


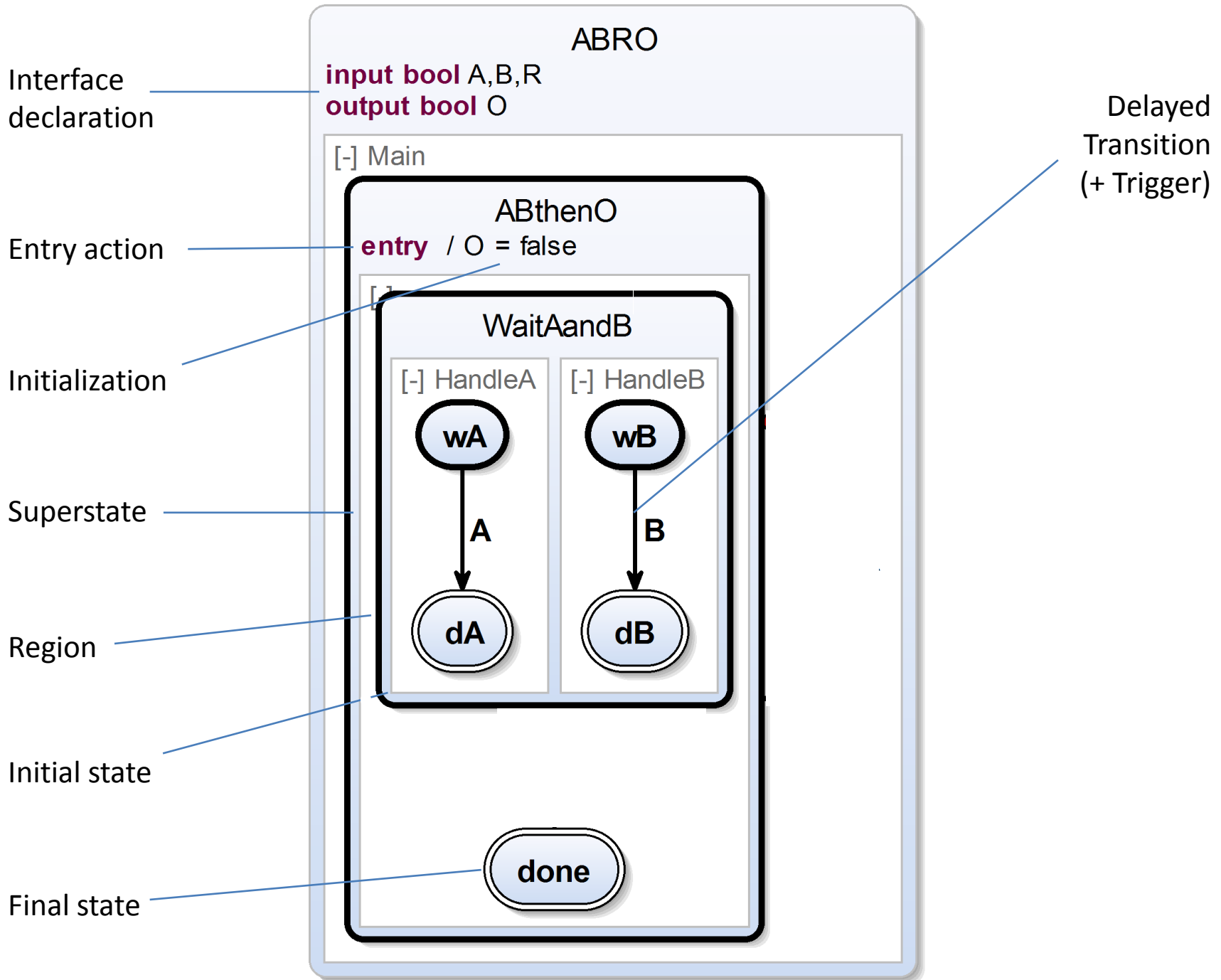


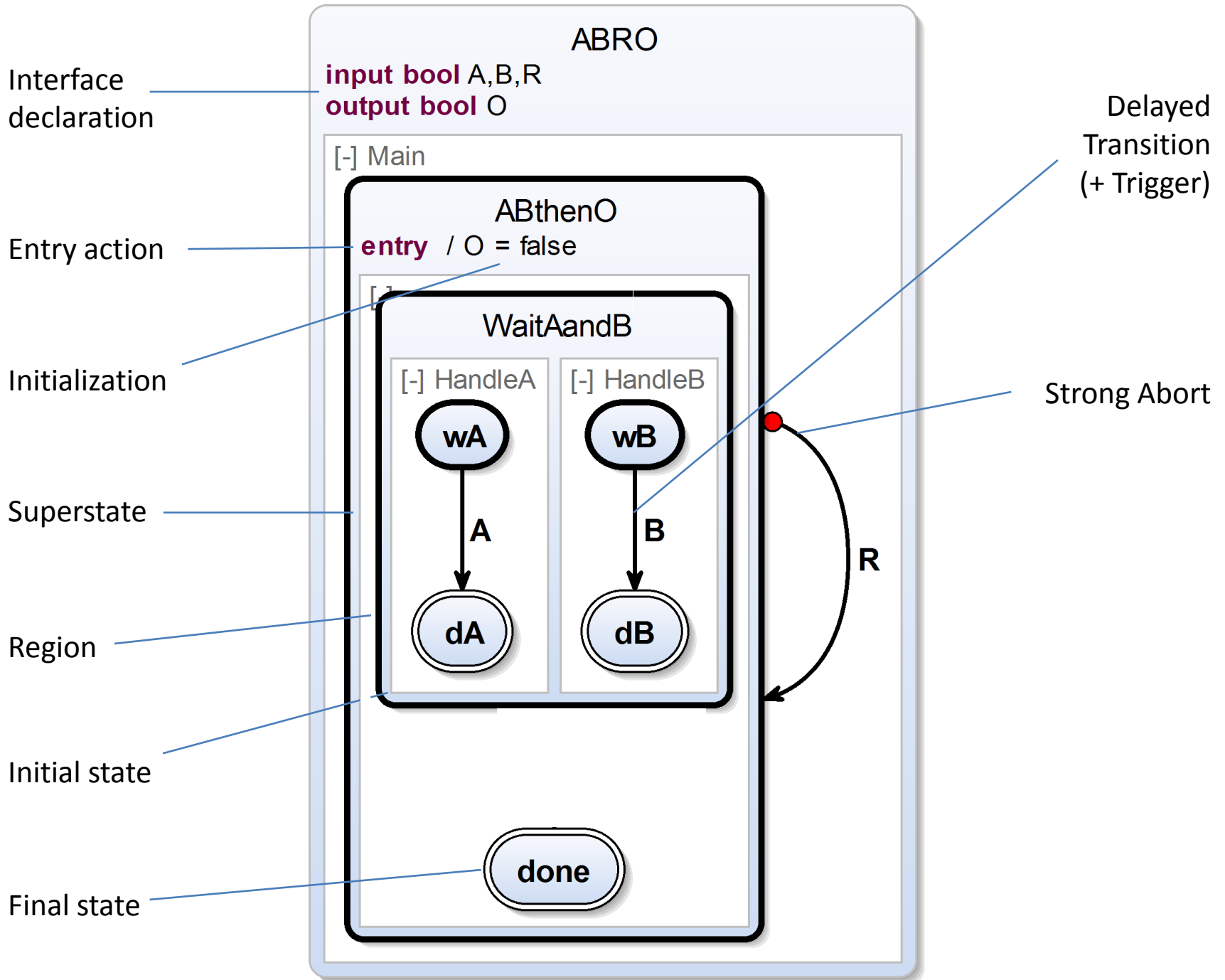


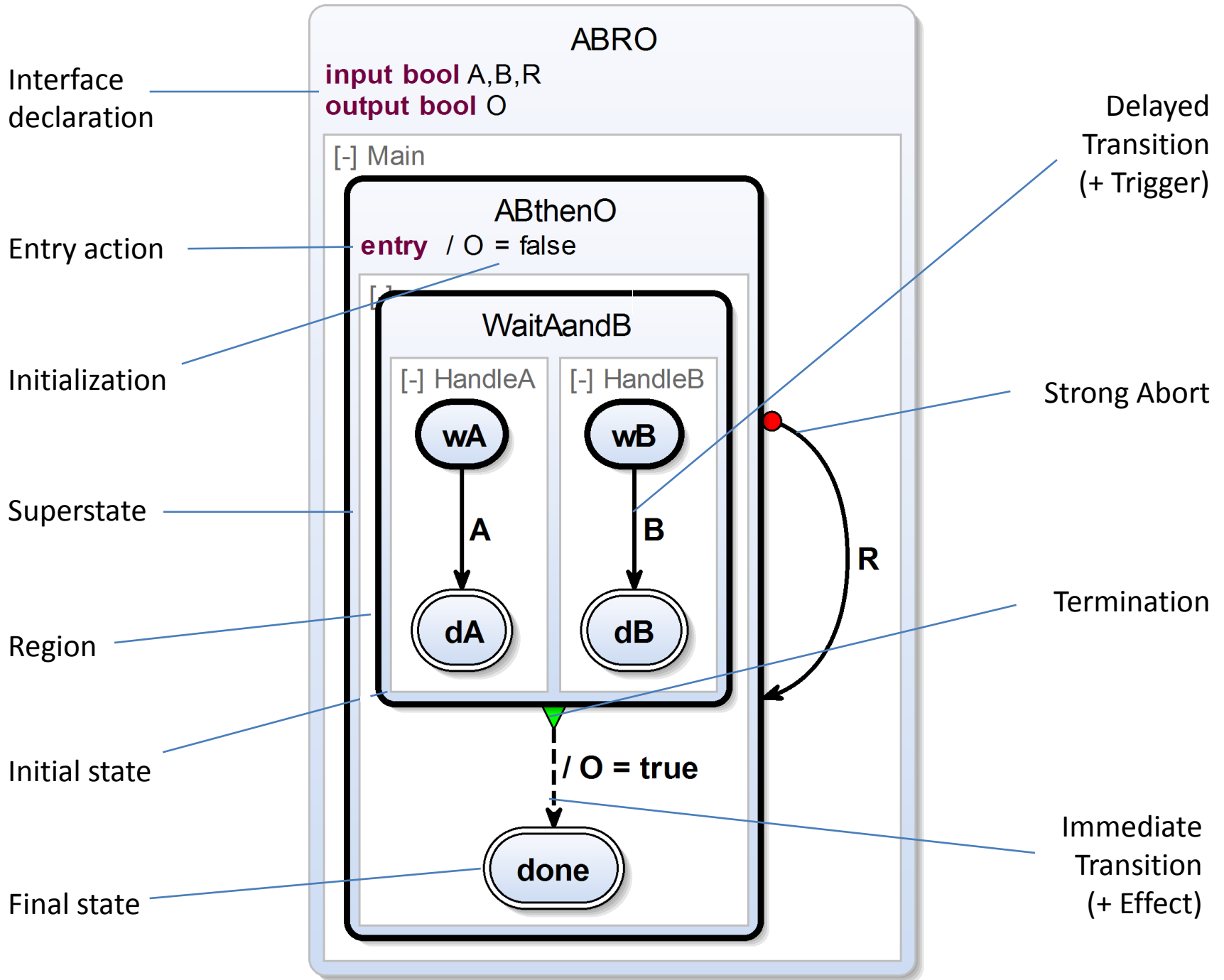






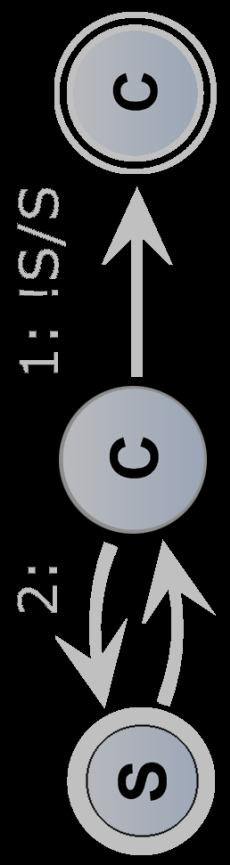
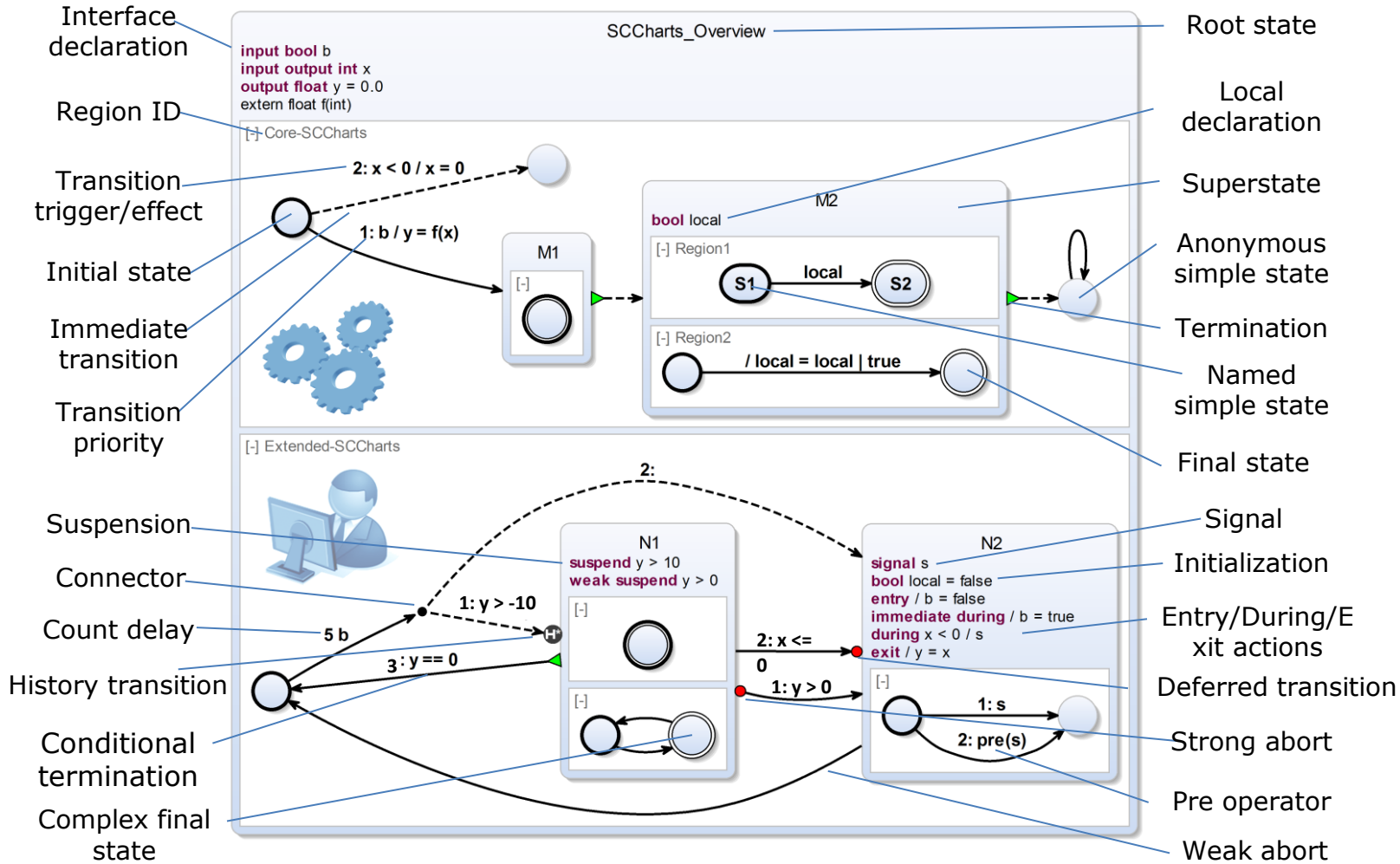








The Whole Menu

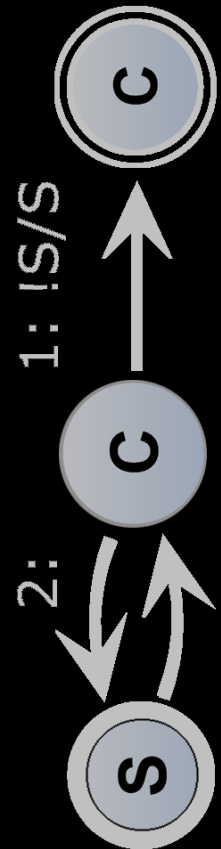


Core-SCCharts
 Small set of simple features ease down stream compilation

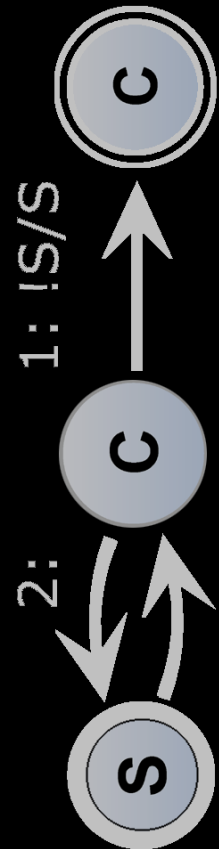
Extended-SCCharts
 Rich set of advanced features ease modeling

SCCharts Ecosystem

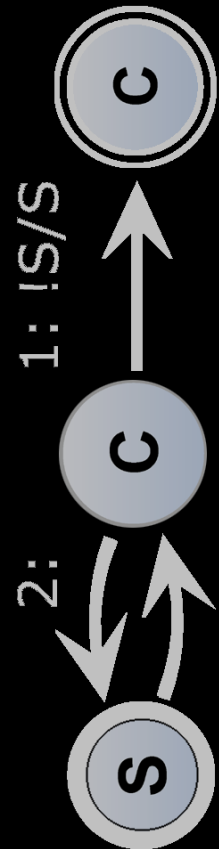
SCCharts
KIELER SCCharts
Mindstorms
Your Feedback



- Papers
- Teaching materials
- On-line compiler
- Question/discussion forum
- Prototype compiler/editor in KIELER



- Starting point:**
 Reinhard von Hanxleden, Björn Duderstadt, Christian Motika, Steven Smyth, Michael Mendler, Joaquín Aguado, Stephen Mercer, Owen O'Brien.
SCCharts: Sequentially Constructive Statecharts for Safety-Critical Applications.
 Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI'14**), Edinburgh, UK, June 2014. ACM.
- Hardware Synthesis:**
 Francesca Rybicki and Steven Smyth and Christian Motika and Alexander Schulz-Rosengarten and Reinhard von Hanxleden.
Interactive Model-Based Compilation Continued – Interactive Incremental Hardware Synthesis for SCCharts.
 Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (**ISoLA 2016**), LNCS, 2016.
- SCCharts and Timing Analysis:**
 Insa Fuhrmann, David Broman, Reinhard von Hanxleden, Alexander Schulz-Rosengarten.
Time for Reactive System Modeling: Interactive Timing Analysis with Hotspot Highlighting.
 Proceedings of the 24th International Conference on Real-Time Networks and Systems (**RTNS'16**), 2016
- SC Esterel:**
 Karsten Rathlev, Steven Smyth, Christian Motika, Reinhard von Hanxleden, Michael Mendler.
SCEst: Sequentially Constructive Esterel.
 Proceedings of the 13th ACM-IEEE International Conference on Formal Methods and Models for System Design (**MEMOCODE'15**), Austin, TX, USA, September 2015.
- Relation to classical synchrony:**
 Joaquín Aguado, Michael Mendler, Reinhard von Hanxleden, Insa Fuhrmann.
Grounding Synchronous Deterministic Concurrency in Sequential Programming.
 Proceedings of the 23rd European Symposium on Programming (**ESOP'14**), LNCS 8410, Grenoble, France, April 2014. Springer.
- Underlying Sequentially Constructive Model of Computation:**
 Reinhard von Hanxleden, Michael Mendler, Joaquín Aguado, Björn Duderstadt, Insa Fuhrmann, Christian Motika, Stephen Mercer, Owen O'Brien, Partha Roop.
Sequentially Constructive Concurrency—A Conservative Extension of the Synchronous Model of Computation.
ACM Transactions on Embedded Computing Systems, Special Issue on Applications of Concurrency to System Design, 13(4s):144:1–144:26, July 2014.
- Foundations:**
 Joaquín Aguado and Michael Mendler and Reinhard von Hanxleden and Insa Fuhrmann.
Denotational Fixed-Point Semantics for Constructive Scheduling of Synchronous Concurrency.
Acta Informatica, Special Issue on Combining Compositionality and Concurrency, 52(4):393–442, 2015.



https://www.../lectures/ x +

https://www.informatik.uni-kiel.de/~rvh/ss15/v-synch/lectures/ mac screenshot

Synchronous Languages

Lecture Notes
 Real-Time and Embedded Systems Group
 Sommersemester 2015

Lecturer: Reinhard v. Hanxleden

The **slides+notes** version is 4-up, includes notes, and uses a printer-friendly coloring scheme. To save paper, empty notes are eliminated, except when they are necessary to place subsequent (non-empty) notes below the slide to which they belong.
 The **slides** version of the lectures is as shown in class, without animation (only one page per slide).
 The **animated** version of the lectures is as shown in class, including animation (may be multiple pages per slide).
 The bottom of this page contains further [notes](#).

- [Lecture 1, 13 April 2015: Introduction \(posted 09 Jun 2015, 12:13 hrs\)](#)
- [Lecture 2, 15 April 2015: Esterel I - Overview \(posted 09 Jun 2015, 12:24 hrs\)](#)
- [Lecture 3, 29 April 2015: Esterel II - Pragmatics \(posted 09 Jun 2015, 12:25 hrs\)](#)
- [Lecture 4, 6 May 2015: Esterel III - The Logical Semantics \(posted 09 Jun 2015, 12:26 hrs\)](#)
- [Lecture 5, 13 May 2015: Esterel IV - The Constructive Semantics \(posted 09 Jun 2015, 12:27 hrs\)](#)
- [Lecture 6, 18 May 2015: Esterel V - The Constructive Circuit Semantics \(posted 09 Jun 2015, 12:27 hrs\)](#)
- [Lecture 7, 27 May 2015: Schizophrenia Problems \(posted 09 Jun 2015, 12:27 hrs\)](#)
- [Lecture 8, 1 June 2015: Esterel Compilation \(posted 09 Jun 2015, 12:27 hrs\)](#)
- [Lecture 9, 3 June 2015: Analysing Constructiveness \(posted 09 Jun 2015, 12:28 hrs\)](#)
- [Lecture 10, 3 June 2015: SyncCharts \(posted 09 Jun 2015, 12:28 hrs\)](#)
- [Lecture 11, 10 June 2015: SCCharts - Sequentially Constructive Statecharts for Safety-Critical Applications \(posted 17 Jun 2015, 9:12 hrs\)](#)
- [Lecture 12, 15 June 2015: Code Generation for Sequential Constructiveness \(posted 17 Jun 2015, 9:14 hrs\)](#)
- [Lecture 13, 22 June 2015: Sequentially Constructive Concurrency \(posted 01 Jul 2015, 11:57 hrs\)](#)
- [Lecture 14, 1 July 2015: Sequentially Constructive Concurrency in Practice \(posted 06 Jul 2015, 13:02 hrs\)](#)
- [Lecture 15, 6 July: SCEst - Sequentially Constructive Esterel \(posted 06 Jul 2015, 13:03 hrs\)](#)

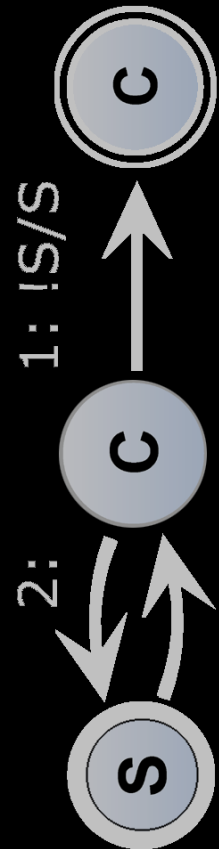
Lecture 1, 13 April 2015: Introduction (slides+notes, slides, animated, posted: 09 Jun 2015, 12:13 hrs)

- About this Class
 - About this class and related classes
 - Practicalities
 - Literature
- Introduction to System Design
 - Embedded and reactive systems
 - Advanced design languages

q isola Hervorheben Groß-/Kleinschreibung Ganze Wörter 1 von 2 Übereinstimmungen

<http://www.rtsys.informatik.uni-kiel.de/en/teaching>

- Slides and exercises available
- Ask rvh for sources



KIELER SCCharts Compil... x +

www.sccharts.com mac screenshot

Textual SCCharts Model

SCT-Language

Load example...

```

1 // The AO example is one of the simpler SCCharts models. It ha
2 // A and a boolean output O. O is initially set to false, whic
3 // SCCharts feature.
4 // AO waits for A becoming true inside state wA. If A becomes
5 // tick other than the initial one, and AO is still inside sta
6 // transition from wA to dA is taken. This also sets the outpu
7 // Note that AO is not responsive in state dA. However, as dA
8 // state, AO does not terminate.
9
10 @HVLayout
11 //@layout[direction] DOWN
12 scchart AO {
13   input bool A;
14   output bool O = false;
15
16   initial state wA
17   --> dA with A / O = true;
18
19   state dA;
20 }
21

```

Compile Transformations / Output

Show/Hide Advanced Mode

Modeled SCChart Diagram (default)

AO

input bool A
output bool O = false

[-]

wA → dA (A / O = true)

Other Options

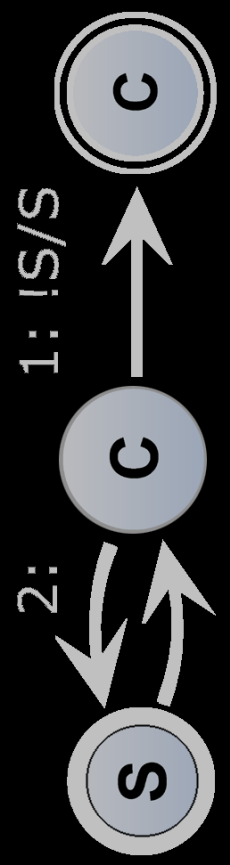
- Textual output (compile only)
- Graphical output (SVG) (compile and render)
- Graphical output (PNG) (compile and render)

Image Quality: 1, Size: fit screen

isola Hervorheben Groß-/Kleinschreibung Ganze Wörter 1 von 2 Übereinstimmungen

www.sccharts.com

Synthesizes SCCharts, circuits, C-code, ...



Questions about "sc..." x Einstellungen x +

https://rtsys.informatik.uni-kiel.de/confluence/questions/topics/12582920/sccharts | kieler confluence questions

RealTime and Embedded Systems | Spaces | People | Questions | Create | ...

What do you want to know? Ask question

Questions Topics Experts Statistics

Questions about "sccharts"

Popular **Recent** Unanswered

0 votes **How can I use arrays of inputs with PROM?**
1 answer • Nis Boerge Wechselberg • Jun 14, 2016
sccharts prom

0 votes **Reset superstate with conditional termination**
1 answer • Sebastian Polney • Jan 14, 2016
sccharts kieler layout development

0 votes **Negative Integers**
1 answer • Sebastian Polney • Jan 14, 2016
nxt integer compilation sccharts

Edit Unwatch

WATCHERS (8) Edit

TOP EXPERT Steven Smyth 35

OTHER EXPERTS

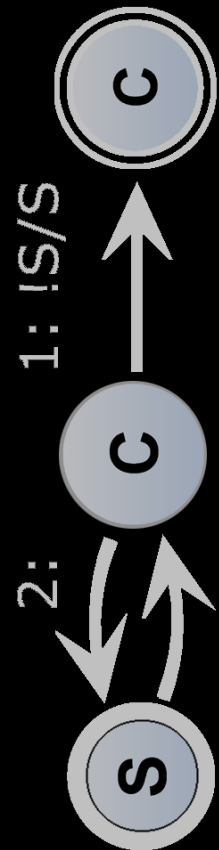
Nis Boerge Wechselberg 10

Felix Eichhorst 10

Q isola | Hervorheben | Groß-/Kleinschreibung | Ganze Wörter | 1 von 2 Übereinstimmungen

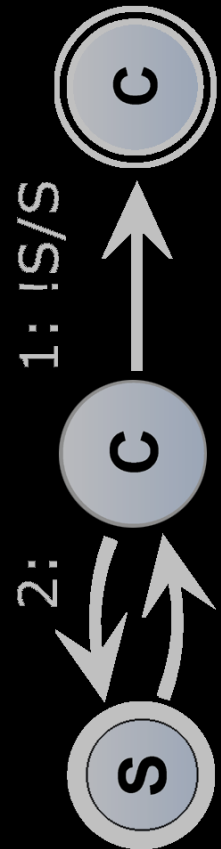
<https://rtsys.informatik.uni-kiel.de/confluence/questions>

SCCharts in KIELER



- Kiel Integrated Environment for Layout Eclipse Rich client
- An academic prototype, for
 - Experimenting with language, synthesis and modeling pragmatics
 - **Teaching synchronous modeling/programming**
- Not industry-strength, but with active support (kieler@informatik.uni-kiel.de)
- Used in classroom since 2013, by about 120 students
- Open source, Eclipse Public License, permits academic and commercial use/extension
- *External users welcome ⇒ this tutorial 😊*

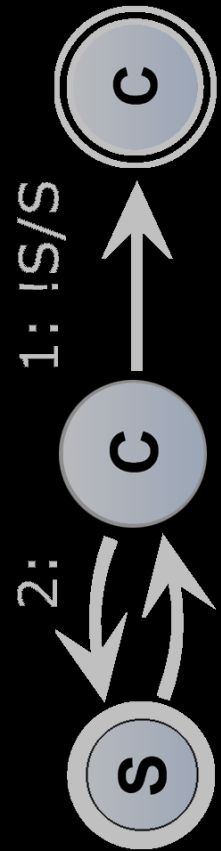
SCCharts Roadmap



Still plenty of things to do:

- Improve code synthesis
- Scheduling variants beyond init-update-read (ForeC, PRET-C, ...)
- Variants on SC MoC (Strict SC, ...)
- Integrate data-flow
- Properly deal with schizophrenia
- Model extraction from legacy code (C)
- Experiment with modeling pragmatics
- Further improve automatic layout
- Wikipedia entry

SCCharts likely to stay around for a while ...



Part II

KIELER SCCharts Tools



The KIELER Project

The Key to Efficient Modeling

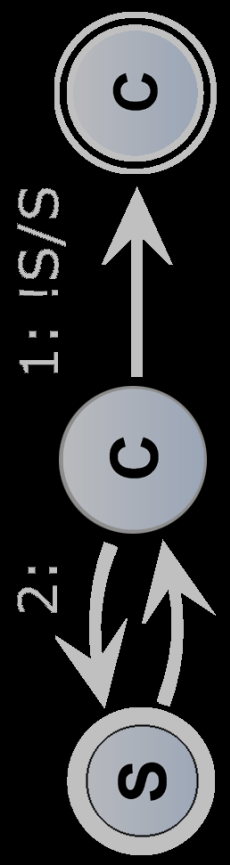


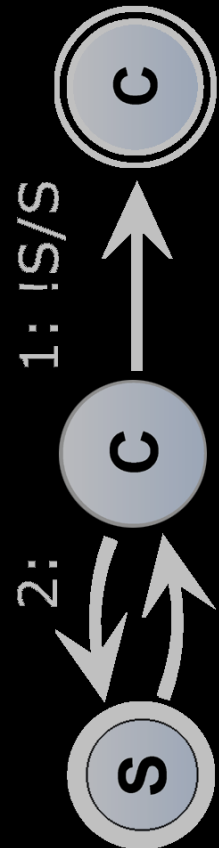
Eclipse Layout Kernel

- Eclipse based
- Since Feb 2008
- 64 committers

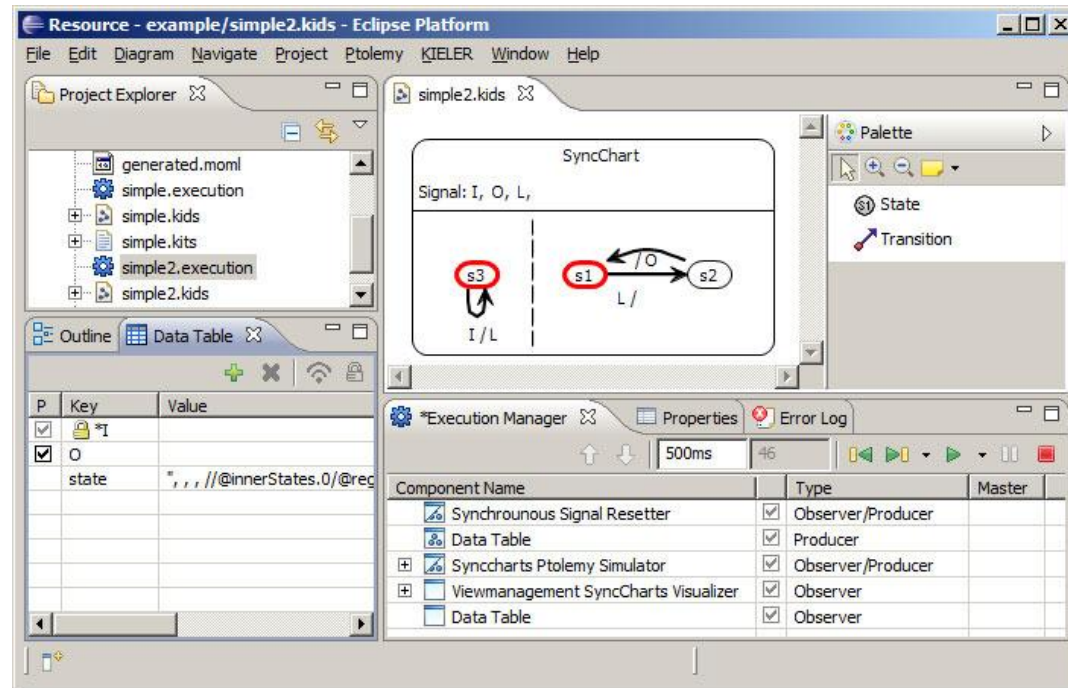
- Pragmatics/Layout + Semantics
- 12,579 commits
- 18,952 files

} Semantics

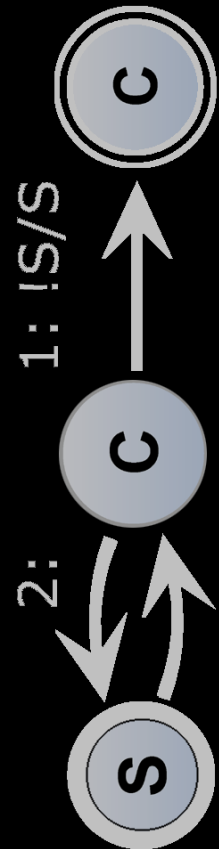




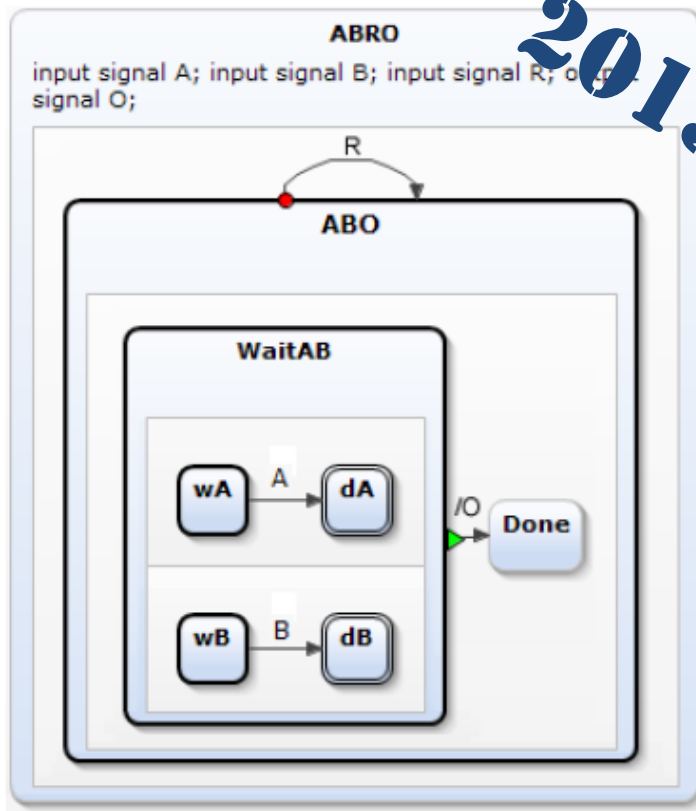
KIELER SCCharts



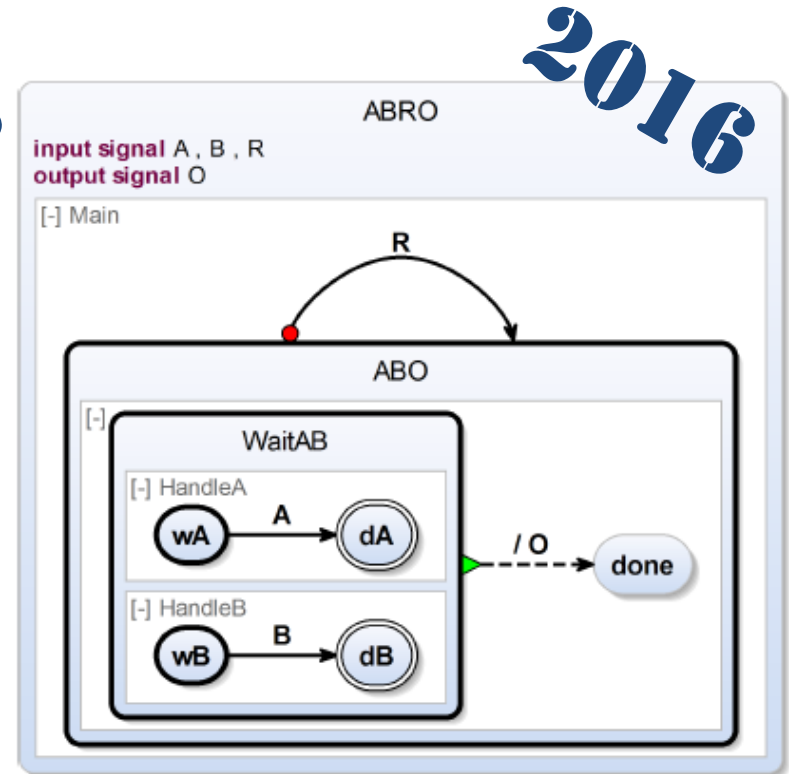
- 2009 Version:**
- Eclipse based modeling (EMF)
 - SyncCharts (subset)
 - **Graphical model editor (GMF)**
 - Generate Ptolemy models
 - Ptolemy simulator



KIELER SCCharts

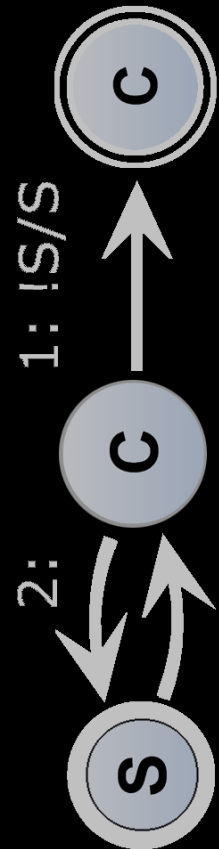


(c) Yakindu SCCharts editor

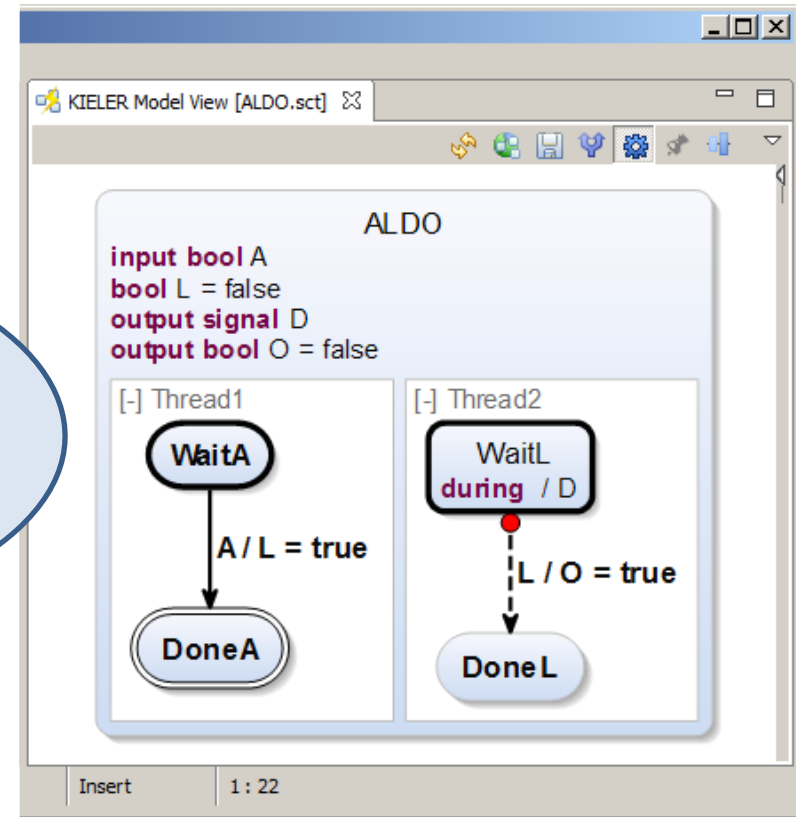


(d) Xtext and KLightD-based SCCharts editor (current)

Textical Modeling

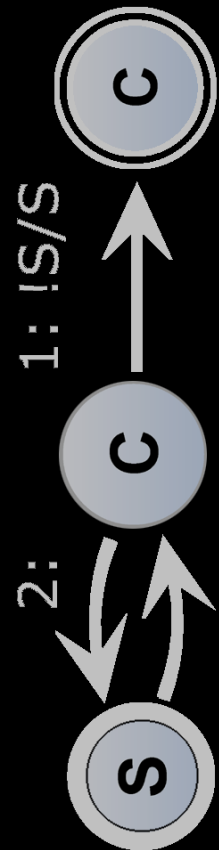


1. Learning Curve
 2. Overview
 3. (Error) Visualizations
 4. Focus & Context
- ...



“Graphical Modeling”

Textual Modeling



 A screenshot of the SCCharts Modeling software interface. The window title is "SCCharts Modeling - Models/ALDO.sct - KIELER". The menu bar includes File, Edit, Navigate, Search, Project, Run, Window, and Help. The main editor displays the code for "ALDO.sct":


```

scchart ALDO "ALDO" {
  input bool A;
  bool L = false;
  output signal D;
  output bool O = false;

  region Thread1:
  initial state WaitA
  --> DoneA with A / L = true;

  final state DoneA;

  region Thread2:
  initial state WaitL {
    during / D
  }
  o-> DoneL immediate with L / O = true;

  state DoneL;
  
```

 The bottom right corner of the editor window has a "Writable" button.

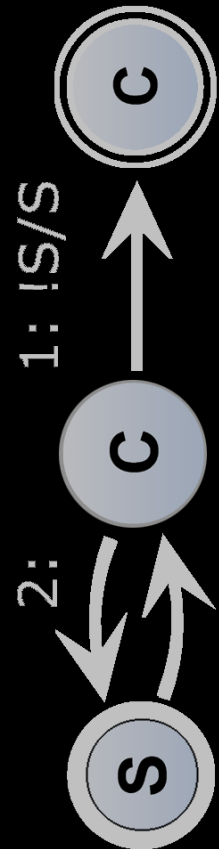
1. Copy & Paste
2. Code Completion
3. Version Control
4. Tool Interchange

...

“Textual Modeling”

Textical Modeling

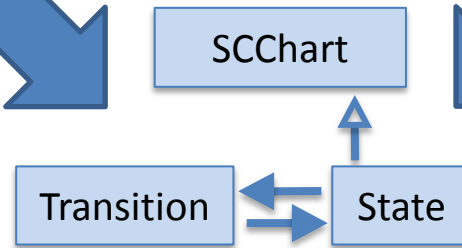
SCCharts
KIELER SCCharts
Mindstorms
Your Feedback



TAKE THE BEST OF BOTH WORLDS

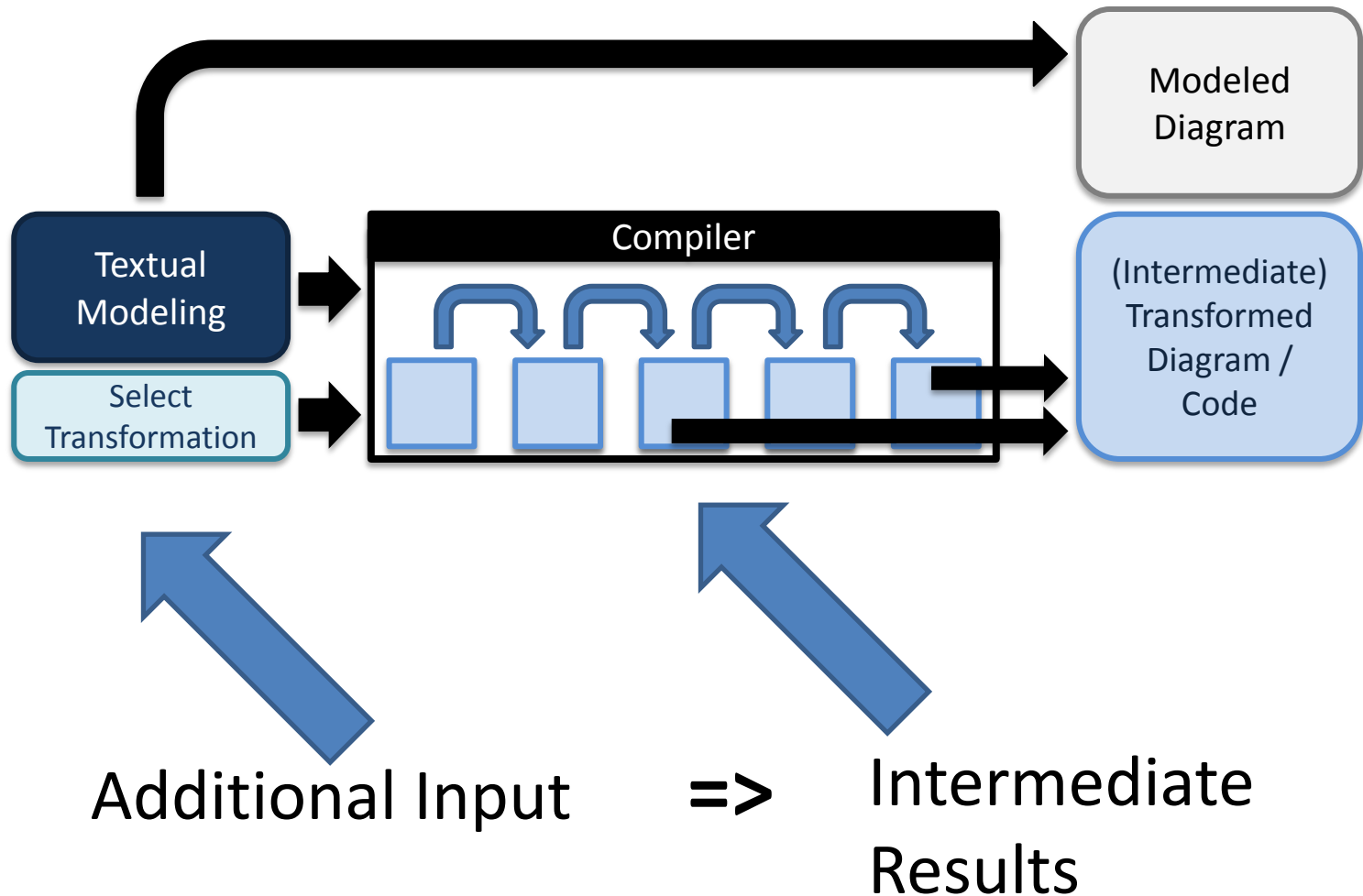
(2) Textual view and editing

(1) Graphical view



(3) Abstract model

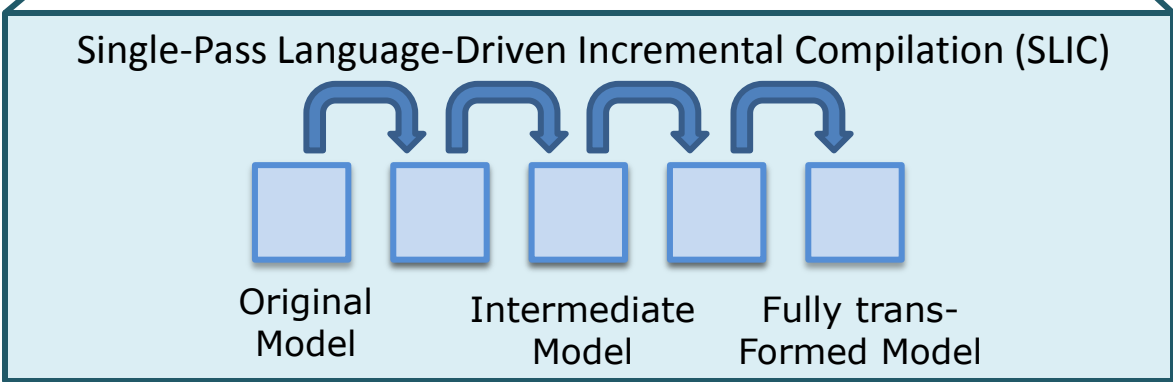
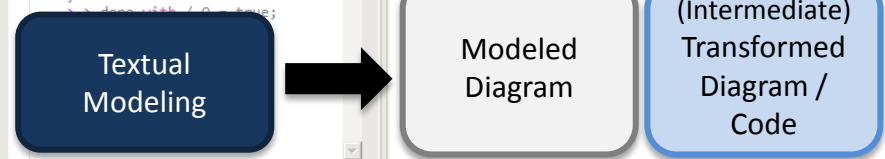
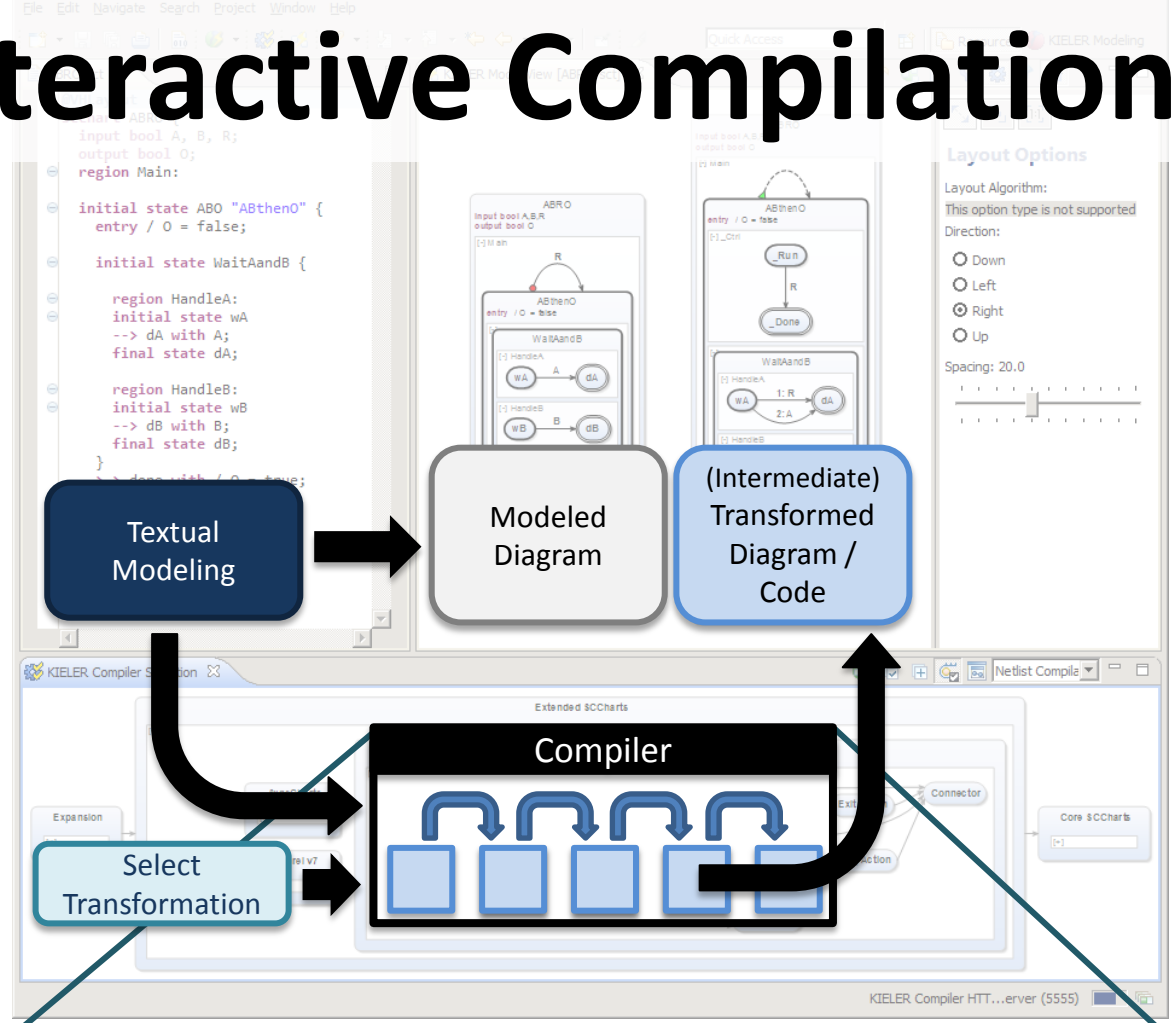
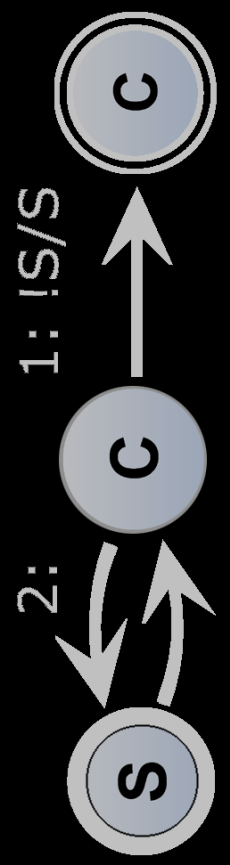
Interactive Compilation



Christian Motika and Steven Smyth and Reinhard von Hanxleden. *Compiling SCCharts — A case-study on interactive model-based compilation*. **ISoLA 2014**, Corfu, Greece, October 2014

Interactive Compilation II

SCCharts
KIELER SCCharts
Mindstorms
Your Feedback



The Tutorial

<http://sccharts.com/tutorial.pdf>

Two parts:

1. KIELER SCCharts
2. Mindstorms

We will synchronize for the second part.

SCCharts Hands-on Tutorial Workshop SYNCHRON'16 Bamberg

1 Tutorial Preparations

Copy the Eclipse-based KIELER SCCharts application from the provided USB thumb drive to your hard drive.

Additionally, you will need a **Java 8 SDK** on your machine (for Windows and the Lego tutorial part you will need the **32bit** version of Java and KIELER SCCharts). You will also find Java 8 installation files on the USB thumb drive in case you need to install it.

The following section, Section 1.1a, describes how to copy the appropriate files using a concrete example build. Note, that the USB stick will contain the most recent build instead.

If you read this tutorial documentation afterwards and are not sitting in the hands-on workshop, then you can download the proper release candidate or the nightly RCA as explained in Section 1.1b.

1.1a Copy from USB Stick

- Plug the provided USB stick: **/USBStick/KIELERSCCharts/**
- Choose a version suited for your OS and copy it to your computer.

1.1b Download

- Use the following URL:
<http://tssy.informatik.uni-kiel.de/~kiel>
or <http://tinyurl.com/zbb2egv>
- Choose a version suited for your OS

www.sccharts.com | SCCharts Tutorial for non-commercial use

SCCharts Cheat Sheet

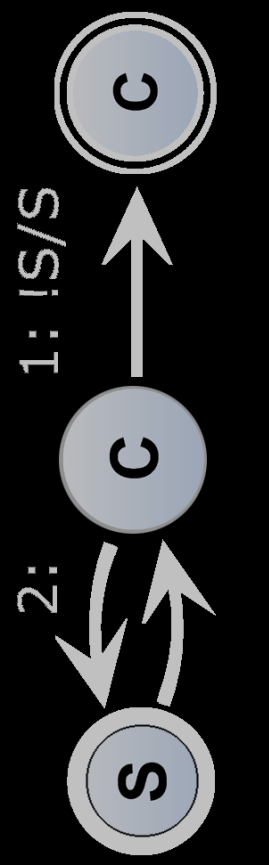
The diagram illustrates various state machine constructs and their notations:

- Initial state A**: `state A { ... }`
- Final state I**: `state I { ... }`
- Transition trigger/effect**: `x <= 0 / x = 0`
- Transition priority**: `1; y >= 10`
- Connector**: `connector state G { ... }`
- Count delay**: `count delay`
- Complex final state**: `final state I { ... }`
- Conditional termination**: `conditional termination`
- Strong abort**: `strong abort`
- Deferred transition**: `deferred transition`
- Suspension**: `suspension`
- Local declaration**: `local bool b;`
- Superstate**: `state n2 { ... region "Region2": ... }`
- Termination**: `termination`
- Anonymous simple state**: `state E { ... }`
- Named simple state**: `state E { ... }`
- Final state**: `final state S2;`
- Signal**: `signal`
- Initialization**: `initialization`
- Entry/During/Exit actions**: `entry / b = false; immediate during / b = true; during x <= 0 / s1; exit / y = x1;`
- Pre operator**: `pre operator`
- Weak abort**: `weak abort`

Legend for state transitions:

- 1: IS/S**: Immediate state transition
- 2:**: Deferred transition
- S**: Strong abort
- C**: Complex final state

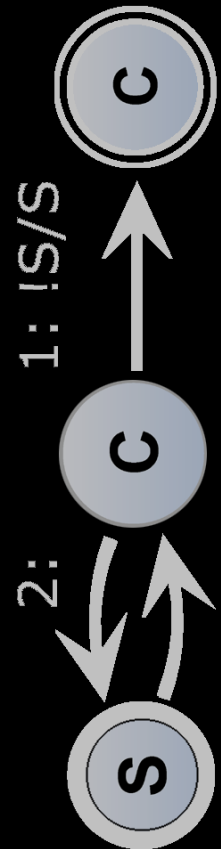
SCCharts Cheat Sheet



Tutorial Preparations



1. Grab a USB thumb drive
2. Copy & extract KIELER SCCharts
3. Start KIELER



Exercise I

Textical Modeling

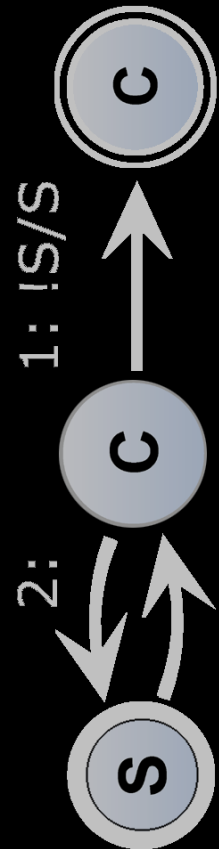
1. KIELER **AO** example (2.1)
2. Modify the example (2.2)
3. SCT editor (2.3)
(formatter, content assist, validation)

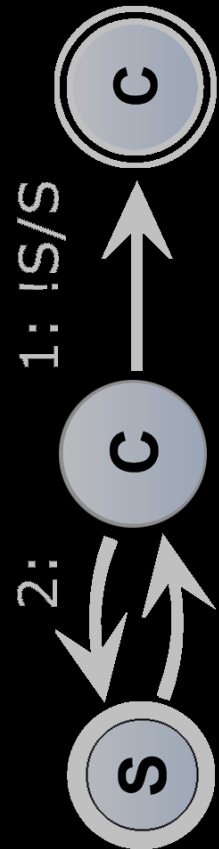


Exercise II

Interactive Compilation

1. Interactive compilation of the **AO** example (3.1)
2. Modify the example in combination with interactive compilation (3.2)
3. Generating code + HW circuits (3.3)

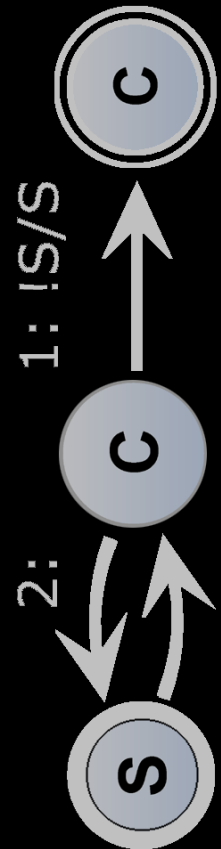




Exercise III

Simulation

1. Simulation Perspective (4.1)
2. Simulation Control (4.2)
3. Synchronous Ticks (4.3)
4. Simulating Intermediate Models (4.4)
5. Hardware Circuit Simulation (4.5)



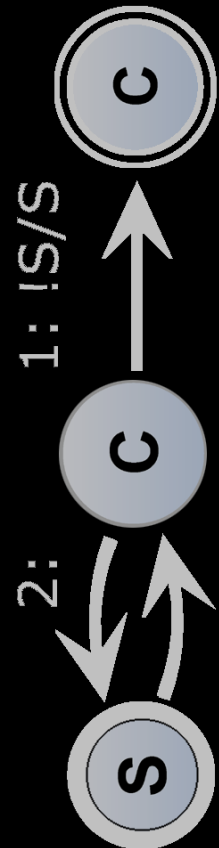
Exercise IV (BONUS)

SCGs

**If time permits,
and you were able to fulfill
the previous tasks faster...**

...then you may want to do this
extra exercise.

**Learn about SCGs and
dependencies in SCCharts**



Tutorial Preparations



1. Grab a USB thumb drive
2. Copy & extract KIELER SCCharts
3. Start KIELER

Exercise I Textual Modeling

1. KIELER **AO** example (2.1)
2. Modify the example (2.2)
3. SCT editor (2.3)
(formatter, content assist, validation)

Exercise II Interactive Compilation

1. Interactive compilation of the **AO** example (3.1)
2. Modify the example in combination with interactive compilation (3.2)
3. Generating code + HW circuits (3.3)

Exercise III Simulation

1. Simulation Perspective (4.1)
2. Simulation Control (4.2)
3. Synchronous Ticks (4.3)
4. Simulating Intermediate Models (4.4)
5. Hardware Circuit Simulation (4.5)

+ Exercise IV (BONUS) **SCGs**

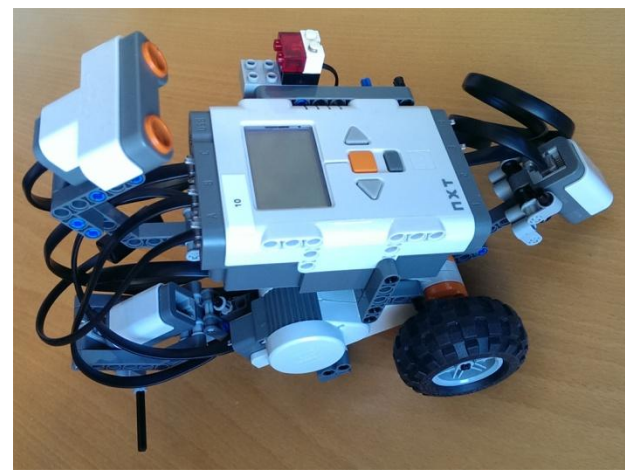
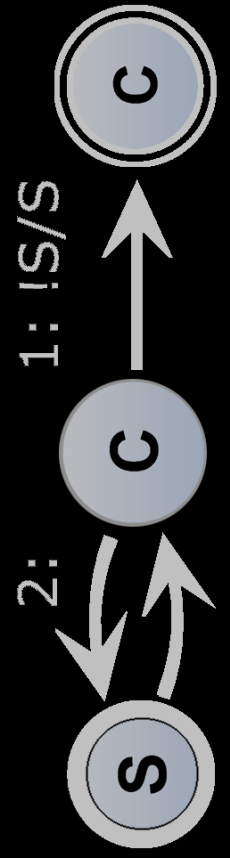
**PROBLEMS?
ASK US!**

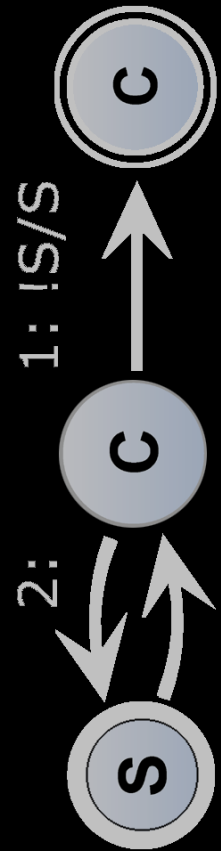


Please finish exercises for now ...

5 MIN LEFT

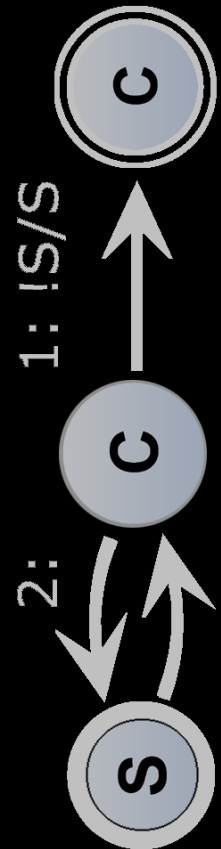
Lego Mindstorms are awaiting you!



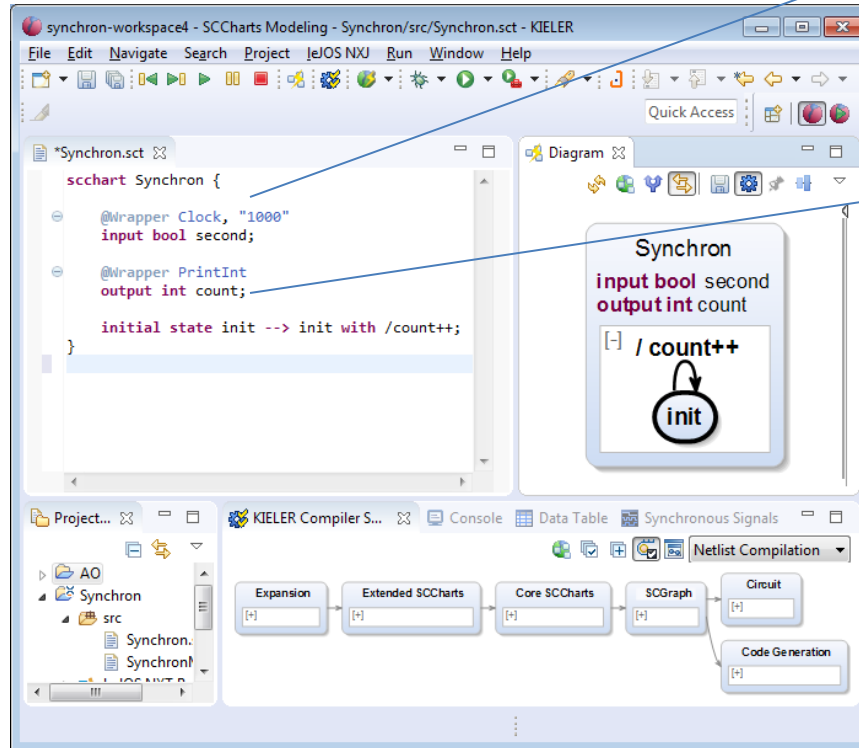


Part III

Lego Mindstorms



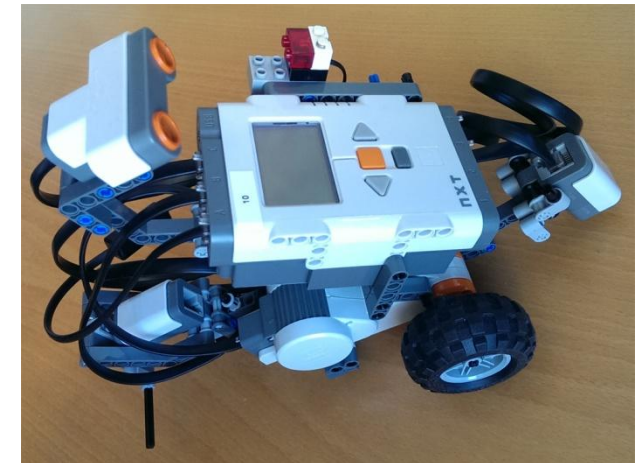
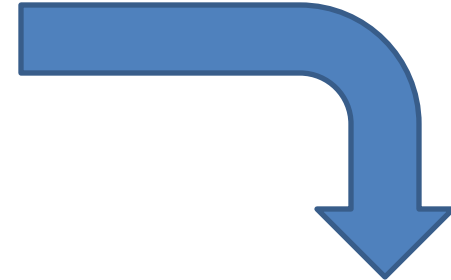
SCCharts & Mindstorms



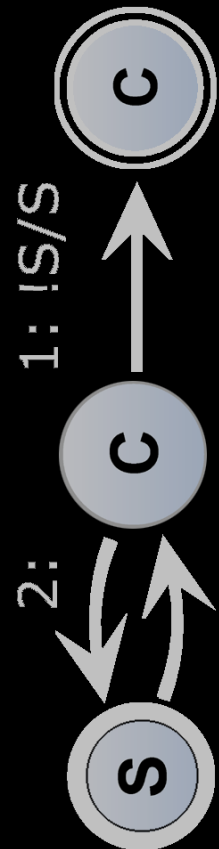
```

@Wrapper Clock, "1000"
input bool second;

@Wrapper PrintInt
output int count;
  
```



Environment Inputs/
 Output Annotations



The Robots

**THANK
YOU**

Additional Supporters

Astrid Flohr

Daniel Grevesmühl

Michael Cyruk

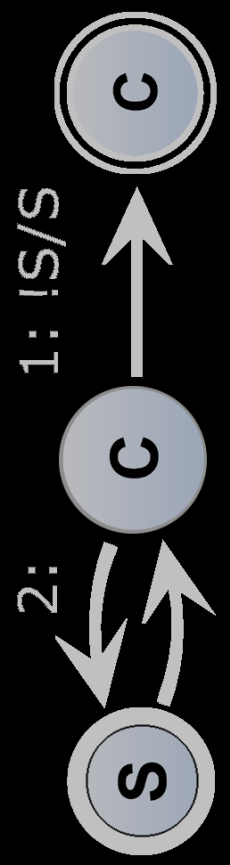
Nelson Tavares
de Sousa

Francesca Rybicki



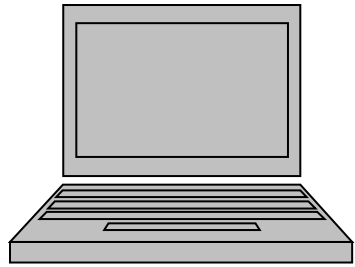
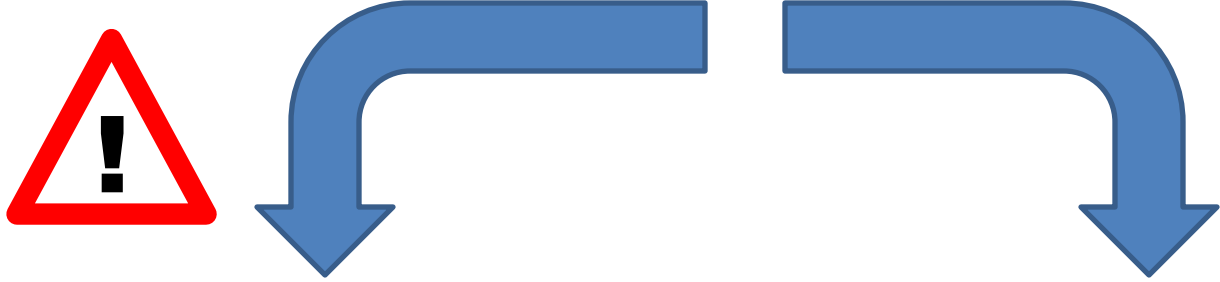
- + Andreas Stange helped with the Environment Snippet Project (PROM)
- + Nis Wechselberg helped with technical issues (Lego)
- + Carsten Sprung implemented the Incremental Update (Diagram View)

Downloading Programs



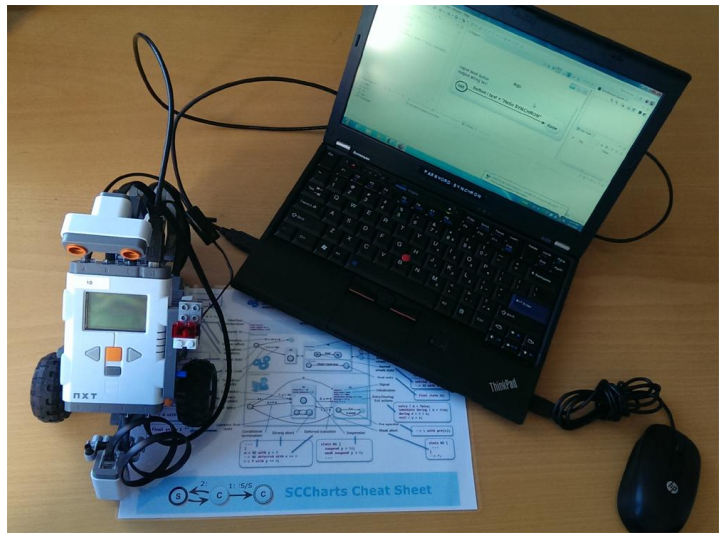
more convenient uploading

no additional installations

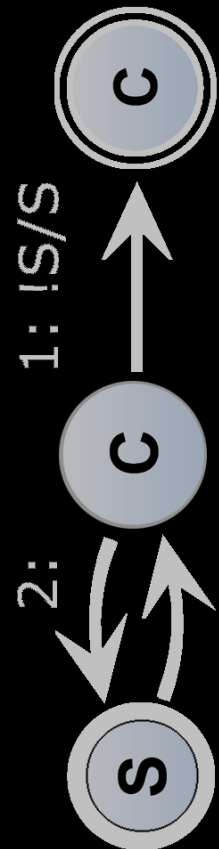


- leJOS
- USB driver
- [Java 32]

use your laptop



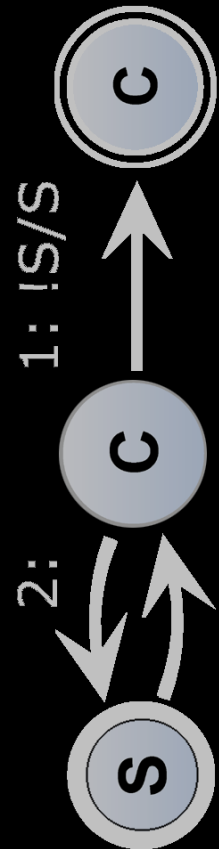
use our
Download Station



Download Station

1. Copy SCT model file to USB stick
2. Insert model into Lego project
3. Possibly adapt model/file names
4. Upload to Mindstorms





Exercise V

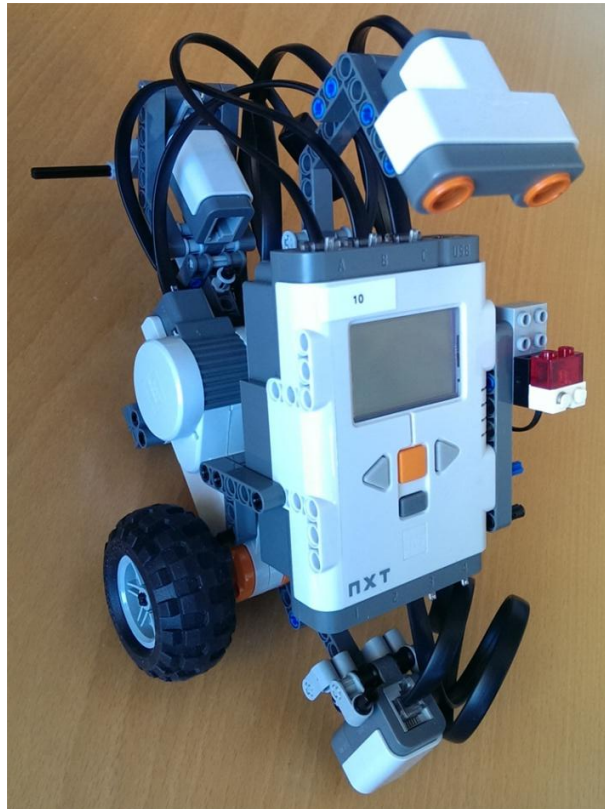
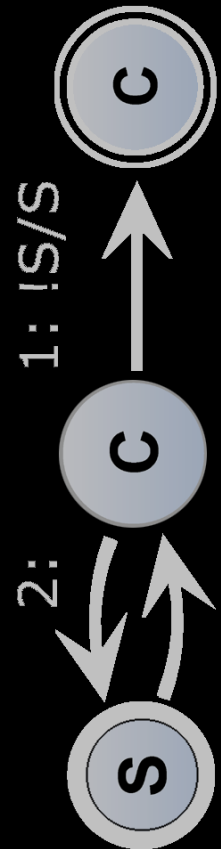
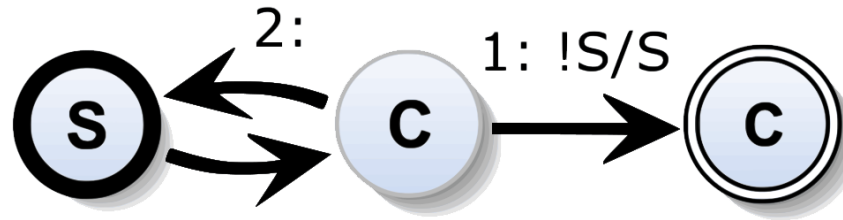
SCCharts & Mindstorms

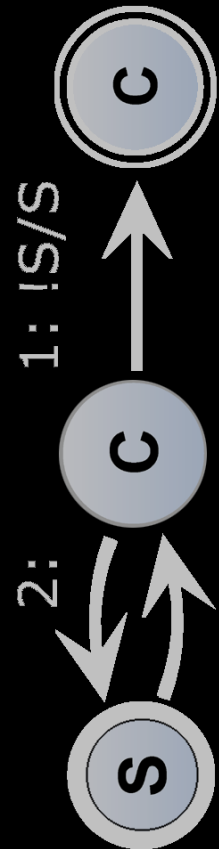
1. Create SCCharts Project (6.1)
2. Modeling & Environment (6.2)
3. Download & Run (6.3)



Exercise VI

SCCharts Path Finder





Your Feedback

SCCharts Hands-on Tutorial Workshop

SYNCHRON'16
Bamberg

SCCHARTS SURVEY FORM

Rev. 1.1t

Have you use KIELER SCCharts before?

Yes
Please let us know in the comments below, where you used it.

No

Can you imagine to use KIELER SCCharts in the future?

Yes in class in projects for fun

No

Rate the quality of the SCCharts development tools you worked with.

Creation/Modeling of models:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

Interactive compilation/code generation:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

Understanding the language semantics:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

Debugging of models:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

Wrapper/Template code generation:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

User interface:

Professional
as other mature open source or commercial products

Advanced
as other smaller commercial or open source products

Ok
as beta version of commercial software or Freeware

Hardly usable
like alpha versions or private/obsolete projects

Comments on KIELER SCCharts:
(E.g., what do you think is missing for the release of the KIELER SCCharts development tools?)

Comments on this tutorial:
(E.g., what can we do to improve it? Was it helpful? Did you enjoy it?)

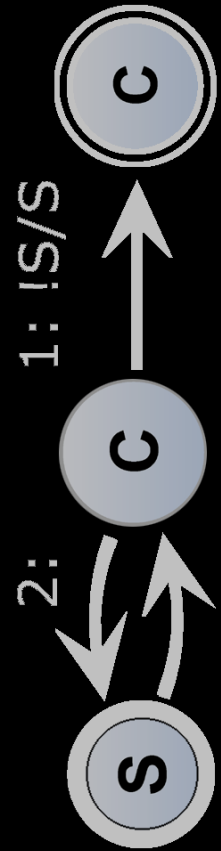
You can use the back page of this form to leave more detailed comments and/or suggestions.
We appreciate your feedback! Thank you very much!

www.sccharts.com | SCCharts Tutorial for non-commercial use only. Do not distribute. Copyright 2016, Real-Time Group, Kiel University, Germany. [34]

... is very
important
for us!

SCCharts
KIELER SCCharts
Mindstorms
Your Feedback

... Now Have Fun! 😊



synchon-workspace4 - SCCharts Modeling - Synchron/src/Synchron.sct - KIELER

```

schart Synchron "SCCharts Hands-on Tutorial" {
  output bool haveFun = false;
  output bool haveMoreFun = false;

  initial state Arrive
  --> R;

  state R "Recall\nSCCharts"
  --> D;

  state D "Use\nSCCharts" {
    during / haveFun = true;
  }
  --> M;

  state M "Program\nMindstorms"
  --> M with / haveMoreFun = true;
}

```

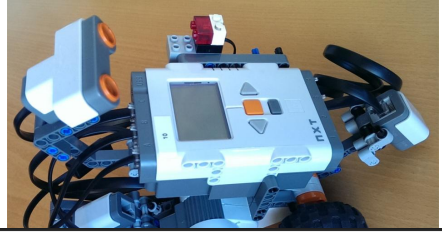
Diagram: SCCharts Hands-on Tutorial

```

graph LR
  Arrive((Arrive)) --> Recall[Recall SCCharts]
  Recall --> Use[Use SCCharts]
  Use --> Program[Program Mindstorms]
  Program --> Use
  
```

Project Explorer: Synchron, SynchronMain.ftl, LeIOS NXT Runtime, kieler-gen, snippets

Netlist Compiler: Expansion, Extended SCCharts, Core SCCharts, SCGraph, Circuit, Code Generation



SCCharts Hands-on Tutorial Workshop

SYNCHRON'16 Bamberg

Rev. 1.1t

SCCHARTS SURVEY FORM

Have you used KIELER SCCharts before?

Yes No

Please let us know in the comments below, when you used it.

Can you imagine to use KIELER SCCharts in the future?

Yes No

Rate the quality of the SCCharts development tools you worked with:

Creation/Modeling of models:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

Debugging of models:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

Interactive compilation/code generation:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

Wrapper/Template code generation:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

Understanding the language semantics:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

User interface:

- Professional
- as other mature open source or commercial products
- Advanced
- as other smaller commercial or open source products
- OK
- as beta version of commercial software or Freeware
- Hardly usable
- like alpha versions or private/obsolete projects

Comments on KIELER SCCharts:
(E.g., what do you think is missing for the release of the KIELER SCCharts development tools?)

Comments on this tutorial:
(E.g., what can we do to improve it? Was it helpful? Did you enjoy it?)

You can use the back page of this form to leave more detailed comments and/or suggestions.
We appreciate your feedback! Thank you very much!

www.sccharts.com | SCCharts Tutorial for non-commercial use only. Do not distribute. Copyright 2016, Real-Time Group, Kiel University, Germany. [54]

THAT'S ALL FOLKS!



Do not hesitate to ask us for help!